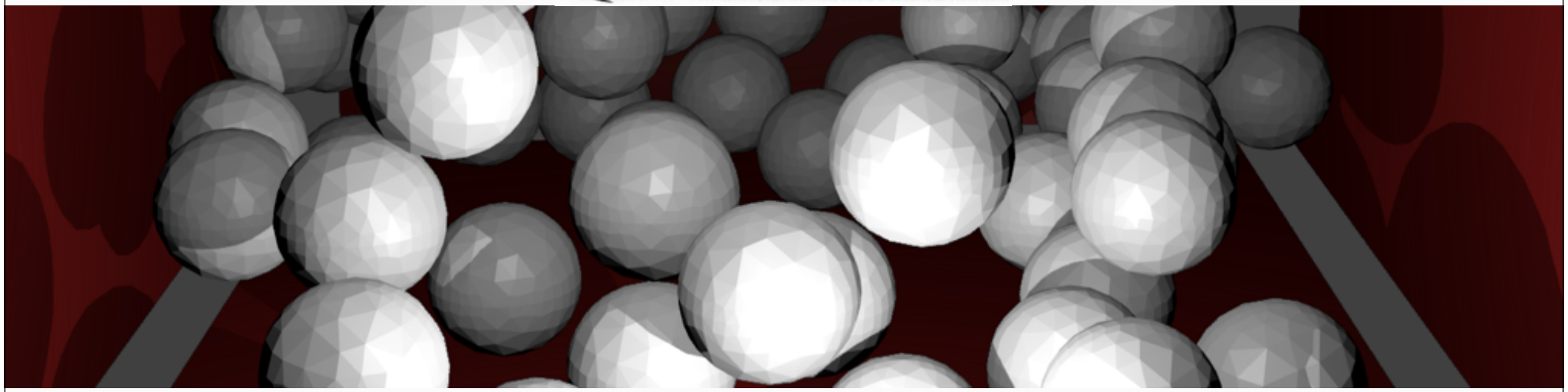


RIGID BODIES SIMULATION

Cristóbal Samaniego

"the devil is in the details"

GENERAL FRAMEWORK

- The **collision detection** refers to the computational problem of estimating the time of contact between the bodies.

- The **Newton-Euler equations** describe the movement of the particles. The bodies move freely until the time of collision is reached.

- Finally, the **collision response** identifies the particles in contact and change their velocities with impulses to prevent interpenetration.

COLLISION DETECTION

Dynamic collision detection. The simulation advance with a dynamic time step.

- The time of collision is estimated for each pair of bodies
- Choose the smallest time.

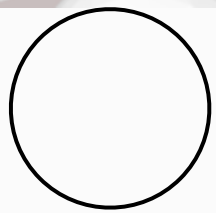
Static collision detection. The simulation advance with a constant time step.

- At each time step, identify the pair of bodies with an intersection
- Correct the penetrations between the bodies

COLLISION DETECTION

Static collision detection. Missing collision

Consider a body at a given time t .

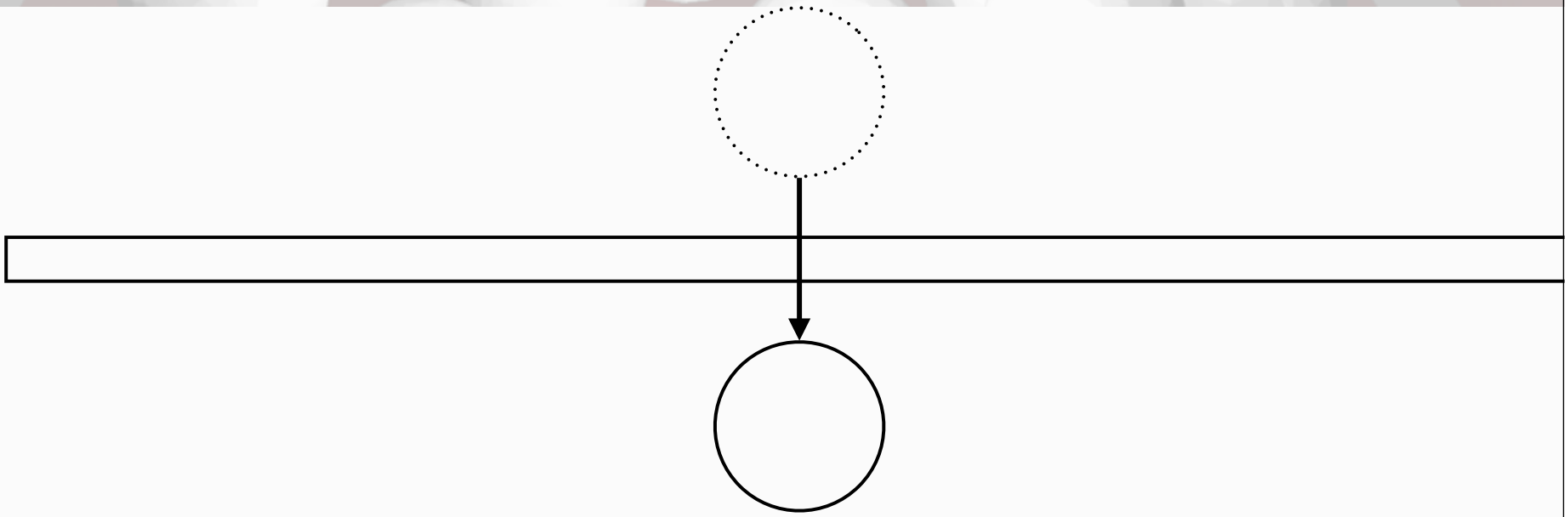


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

COLLISION DETECTION

Static collision detection. Missing collision

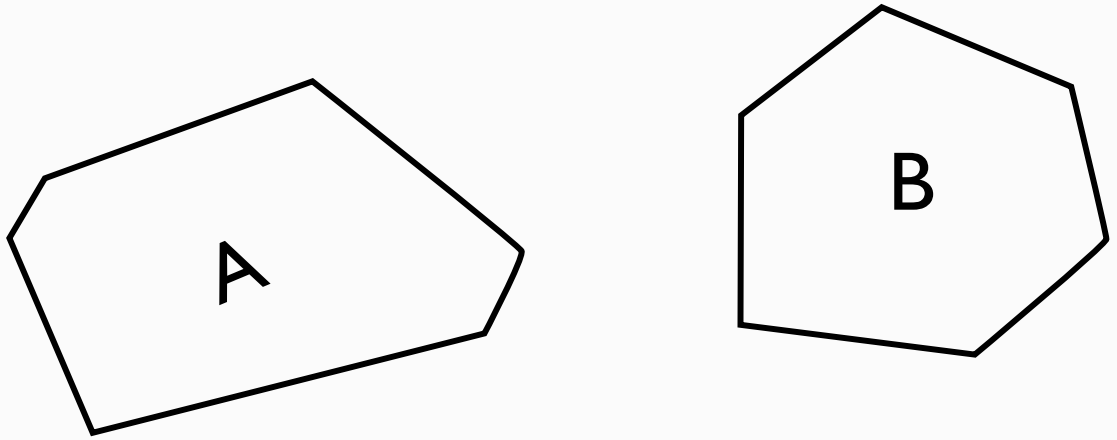
At the next time step $t + \Delta t$.



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

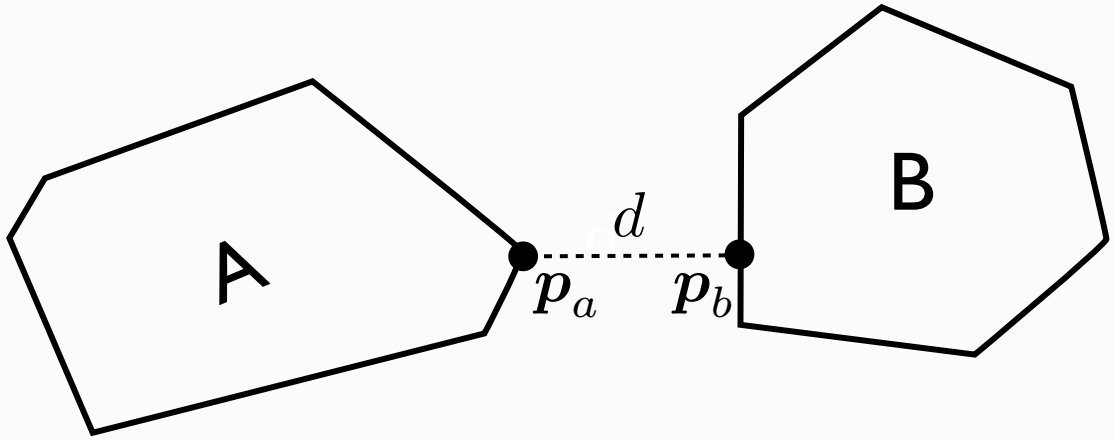
GENERAL FRAMEWORK

Consider two arbitrary convex bodies.



COLLISION DETECTION

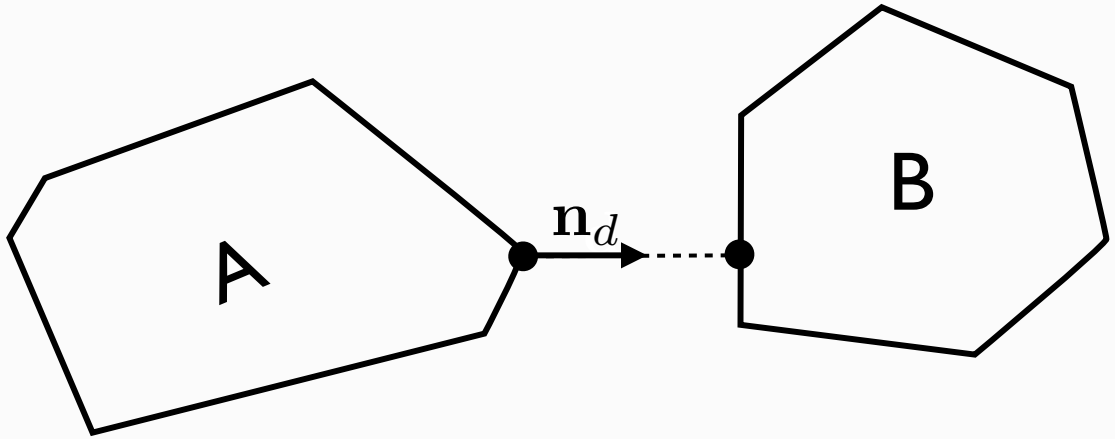
1. Find the closest points between the bodies and its distance d.



COLLISION DETECTION

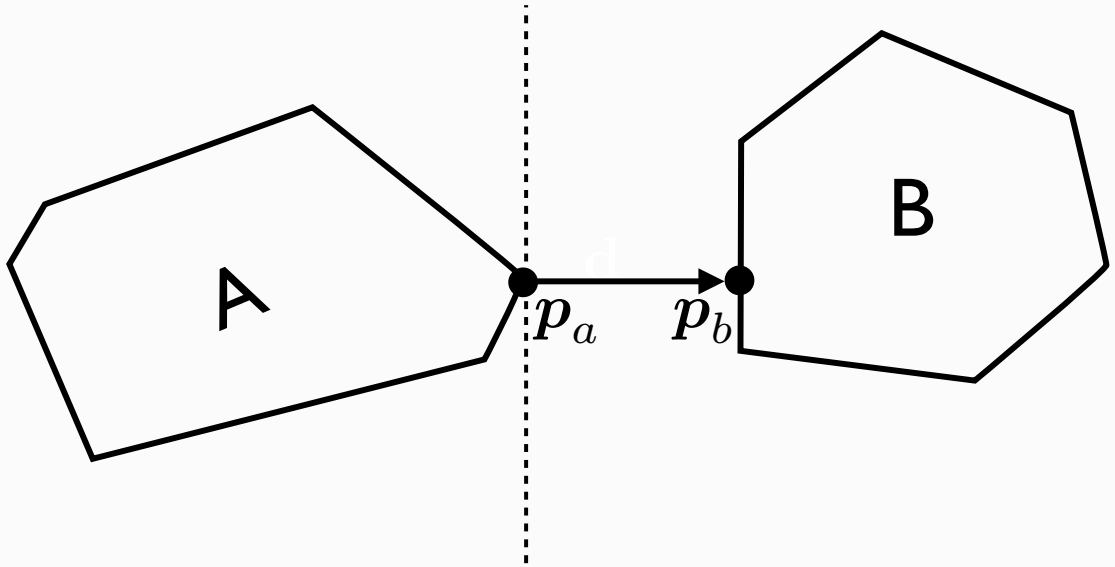
2. Obtain an upper bound on the distance traveled by any point on the body A and B along the unit vector $\mathbf{n}_d$ defined by the closest points:

$$D_A(t), D_B(t).$$



COLLISION DETECTION

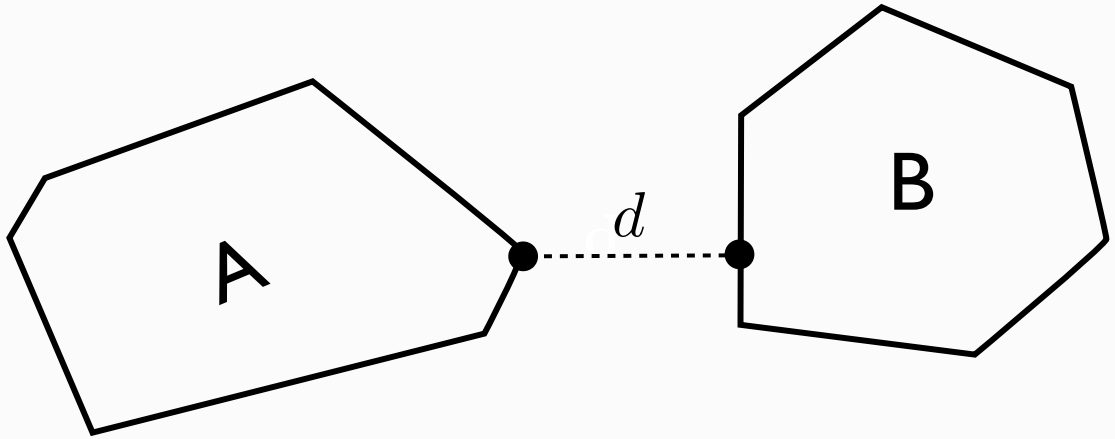
Idea. The plane perpendicular to $\mathbf{n}_d$ that passes through any of the closest points contains entirely the first body on one side and the second body on the other side.



COLLISION DETECTION

3. Finally, estimate the time of collision by solving

$$D_A(t) - D_B(t) = d$$

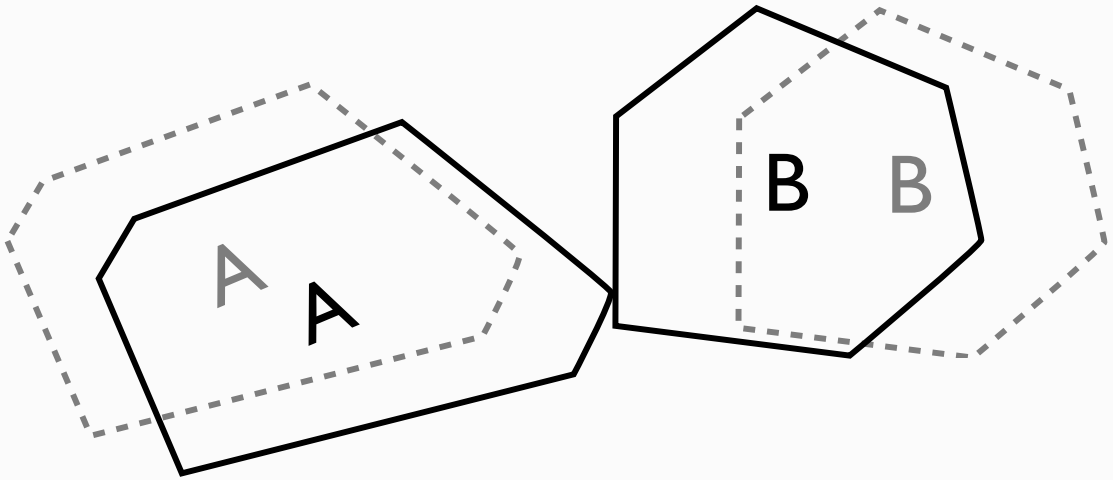


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

RIGID BODIES MOTION

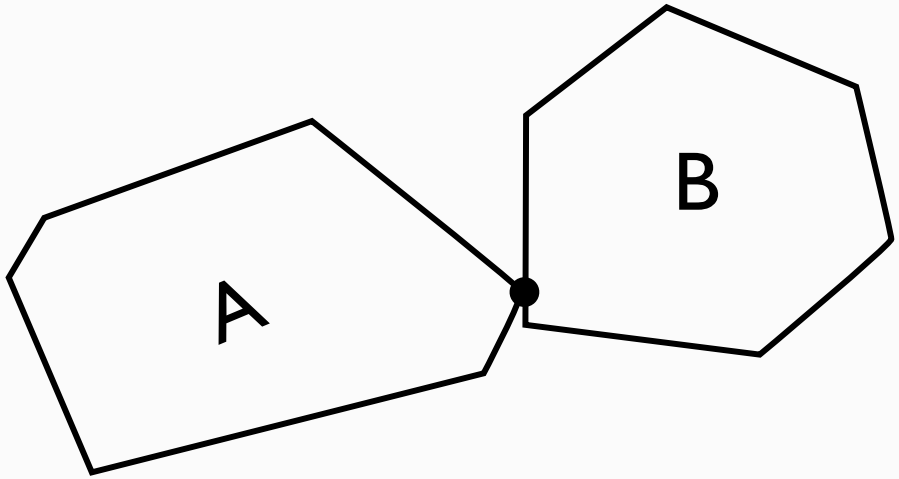
Move the particles until the time of collision is reached solving the Newton-Euler equations: .

$$\mathbf{f} = m\frac{d\mathbf{u}}{dt},\ \boldsymbol{\tau} = \frac{d(\mathbf{I}\cdot\omega)}{dt}$$



COLLISION RESOLUTION

1. Identify if the particles are in contact.

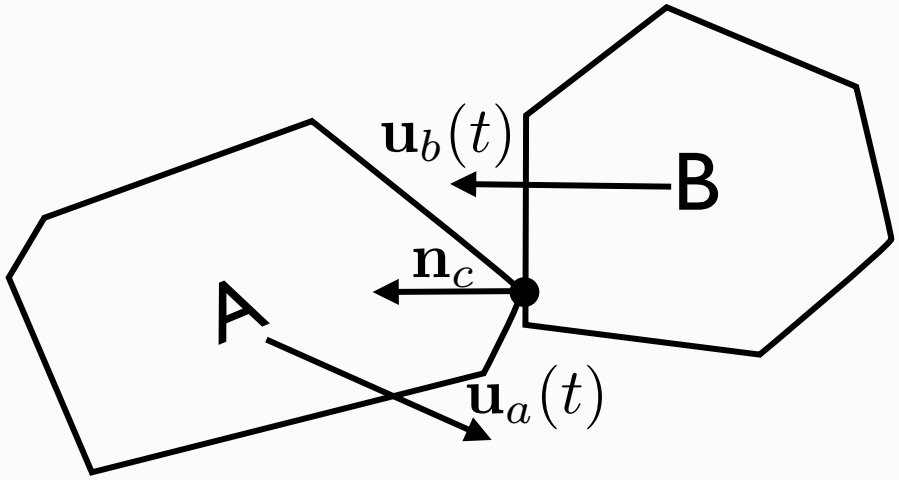


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

COLLISION RESOLUTION

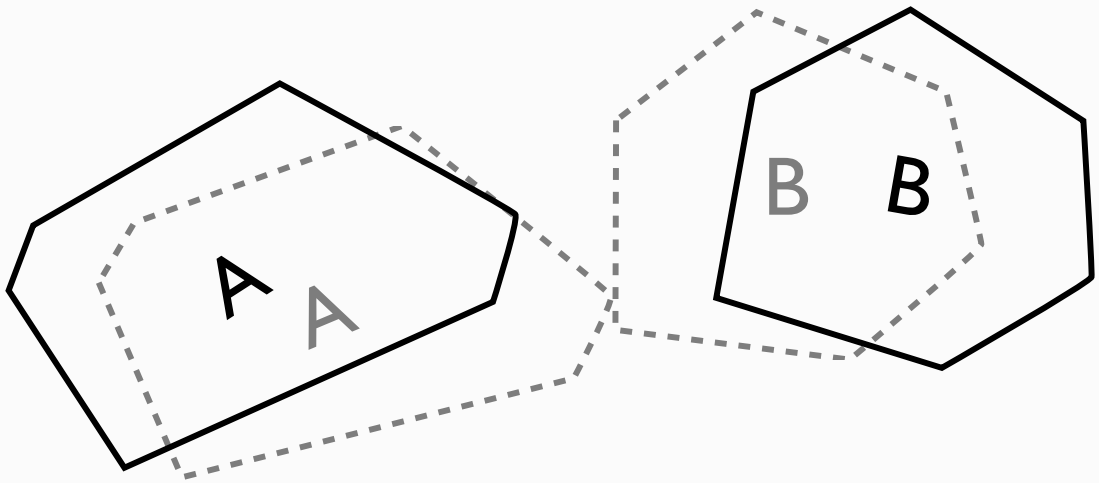
2. Determine the exterior normal of the face of contact $\mathbf{n}_c$ and the relative velocity of the particles:

$$u_{rel} = \mathbf{n}_c \cdot (\mathbf{u}_a(t) - \mathbf{u}_b(t))$$



COLLISION RESOLUTION

3. If $u_{rel} <= 0$ calculate the impulse necessary to prevent interpenetration. An impulse modifies the velocities using the direction of $\mathbf{n}_c$



COLLISION RESOLUTION

Impulse force. Definition

$$\mathbf{J} = \lim_{\Delta t \to 0} \int_{t}^{t+\Delta t} \mathbf{f}(t) dt$$

The impulse can also be expressed as

$$\mathbf{J} = j\underbrace{\mathbf{n}_c}$$

unit exterior normal of the face of contact

COLLISION RESOLUTION

Impulse force. Determine the impulse magnitude considering two bodies A and B in contact

The velocities in body A are related with the previous velocities by equation

$$\mathbf{v}_A^+(t_c) = \mathbf{v}_A^-(t_c) + \frac{j\mathbf{n}(t_c)}{m_A}$$

$$\boldsymbol{\omega}_A^+(t) = \boldsymbol{\omega}_A^-(t) + \mathbf{I}_A^{-1}(t)(\mathbf{r}_A(t_c) \times j\mathbf{n}(t))$$

The empirical law for frictionless collisions says that

$$\mathbf{n}(t) \cdot (\mathbf{u}_A^+(t) - \mathbf{u}_B^+(t)) = -\epsilon\, \mathbf{n}(t_c) \cdot (\mathbf{u}_B^-(t) - \mathbf{u}_B^-(t))$$

FIRST ALGORITHM

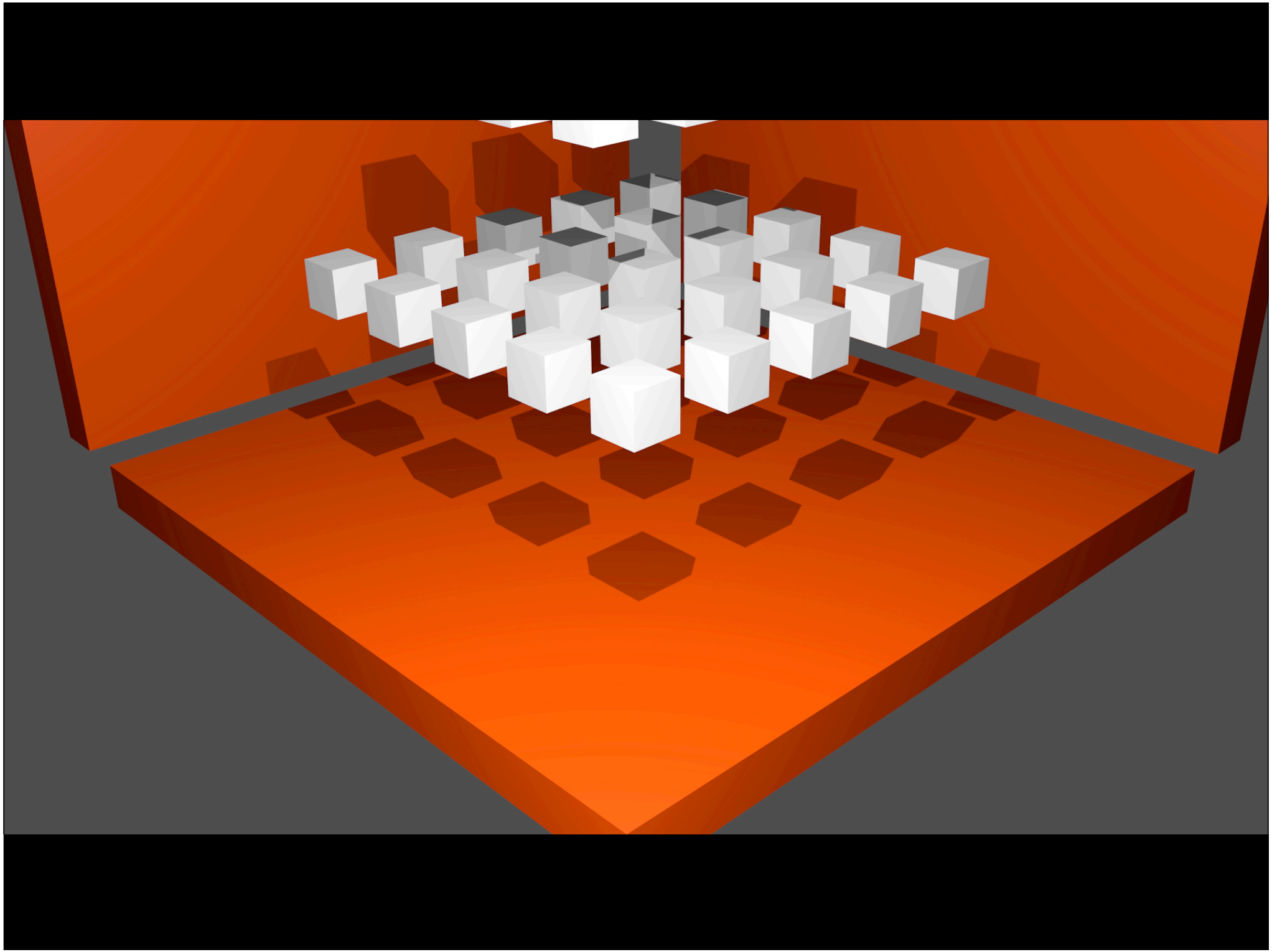
Collision detection

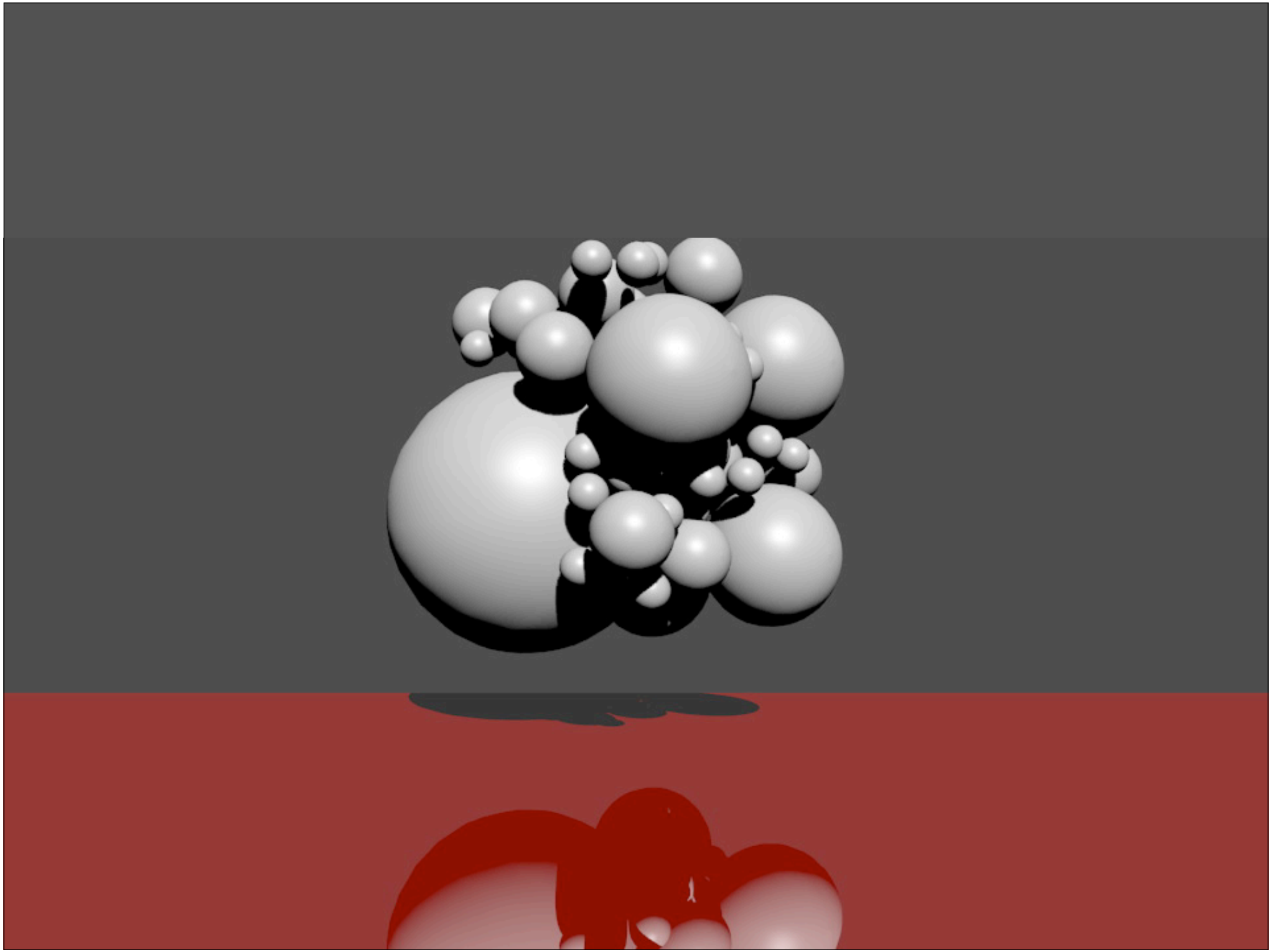
- for each body A in the simulation do
  - for each body B in the simulation different from A do
    - determine the closest points between A and B.
    - determine the distance traveled by A and B along the unit vector defined by their closest points.
    - estimate the time of collision between A and B.
    - if this time is the minimum obtained until now then
      - store the time of collision
    - end if
  - end for
- end for

FIRST ALGORITHM

Collision response

- do while a contact be founded
  - for each body A in the simulation do
    - for each body B in the simulation different from A do
      - if A and B are in contact then
        - determine their relative velocity
        - if their relative velocity is less or equal to zero then
          - determine the impulse to prevent interpenetration
        - end if
      - end if
    - end do
  - end for
- end do

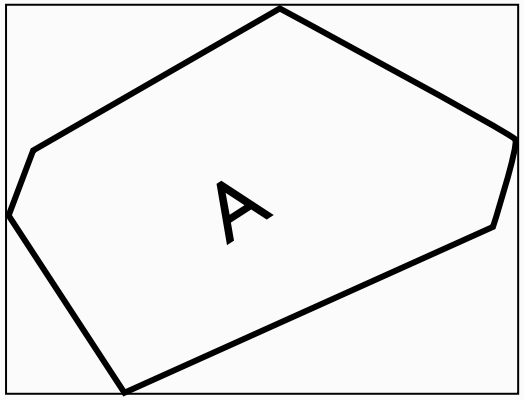


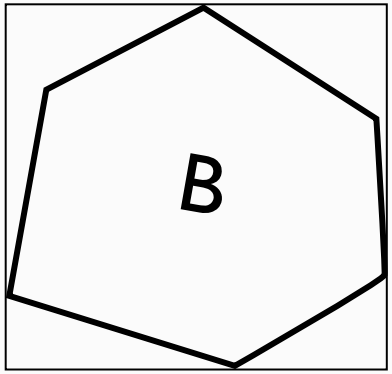


IMPROVEMENTS

Boundary Boxes

A easy way to avoid unnecessary work is to check first if the particle boundary boxes intersect.

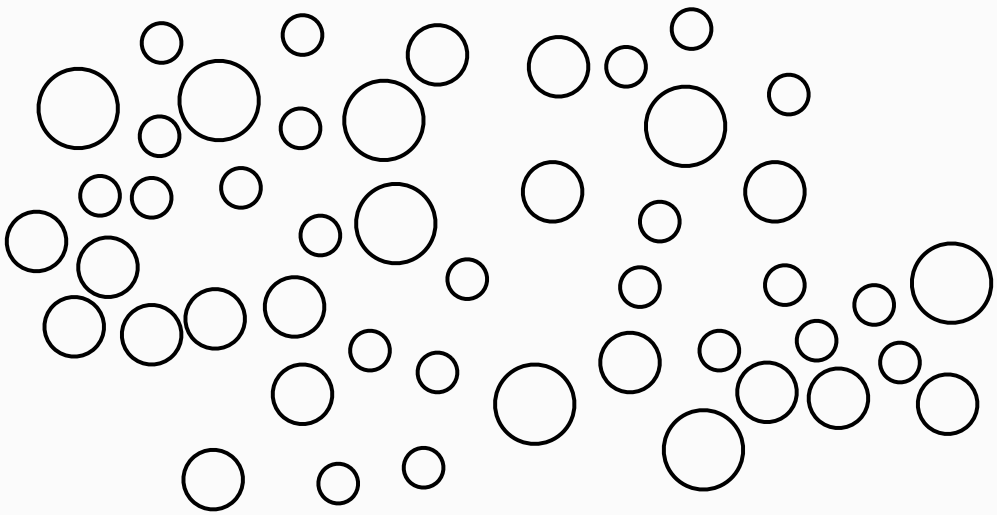




IMPROVEMENTS

Bucket sort

Consider the following group of bodies.

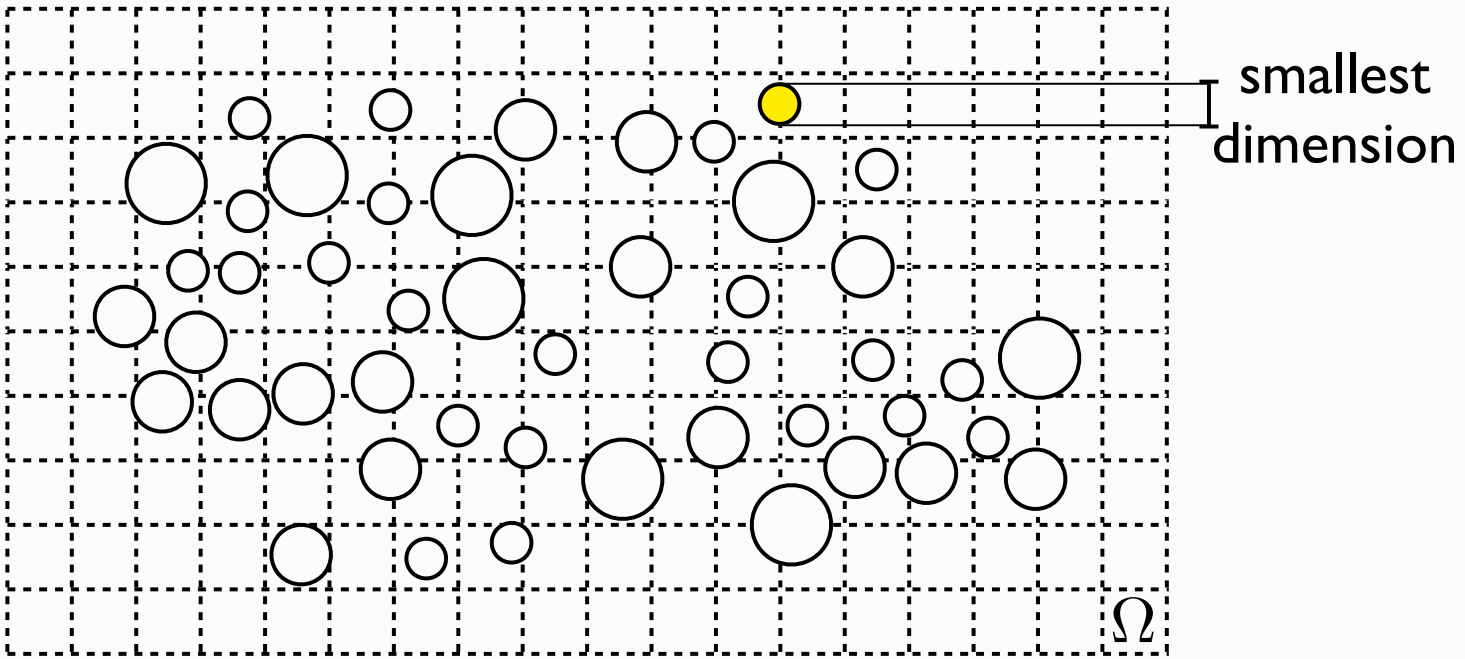


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

IMPROVEMENTS

Bucket sort

1. Subdivide the domain of the problem into square boxes. The size of each box is proportional to the smallest body dimension.

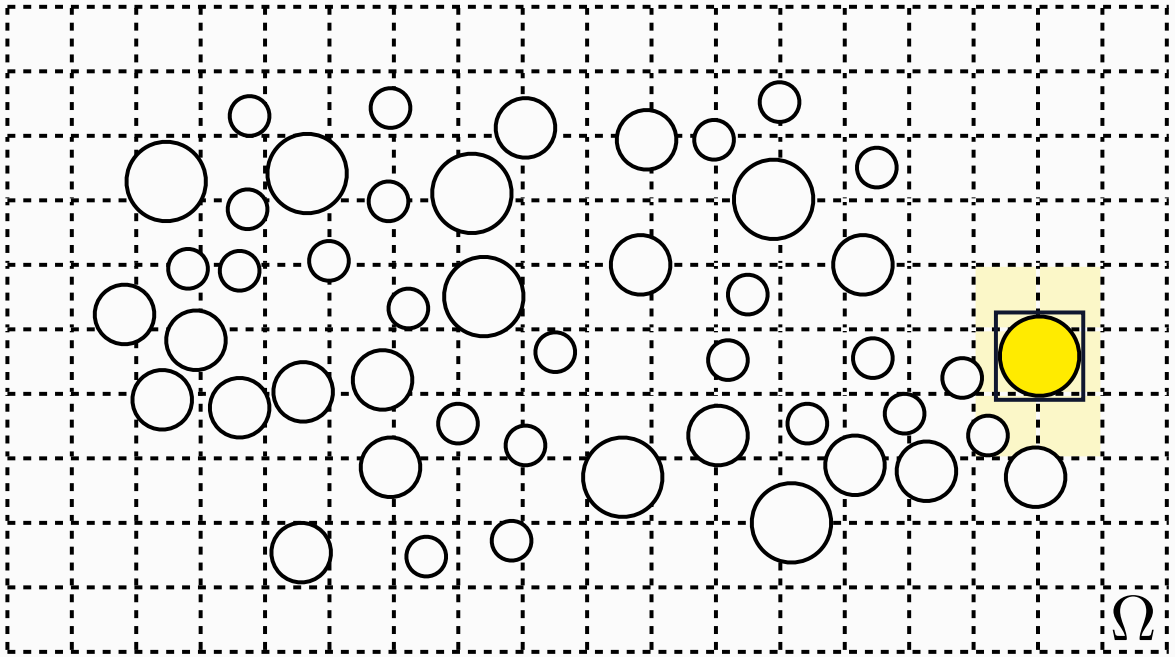


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

IMPROVEMENTS

Bucket sort

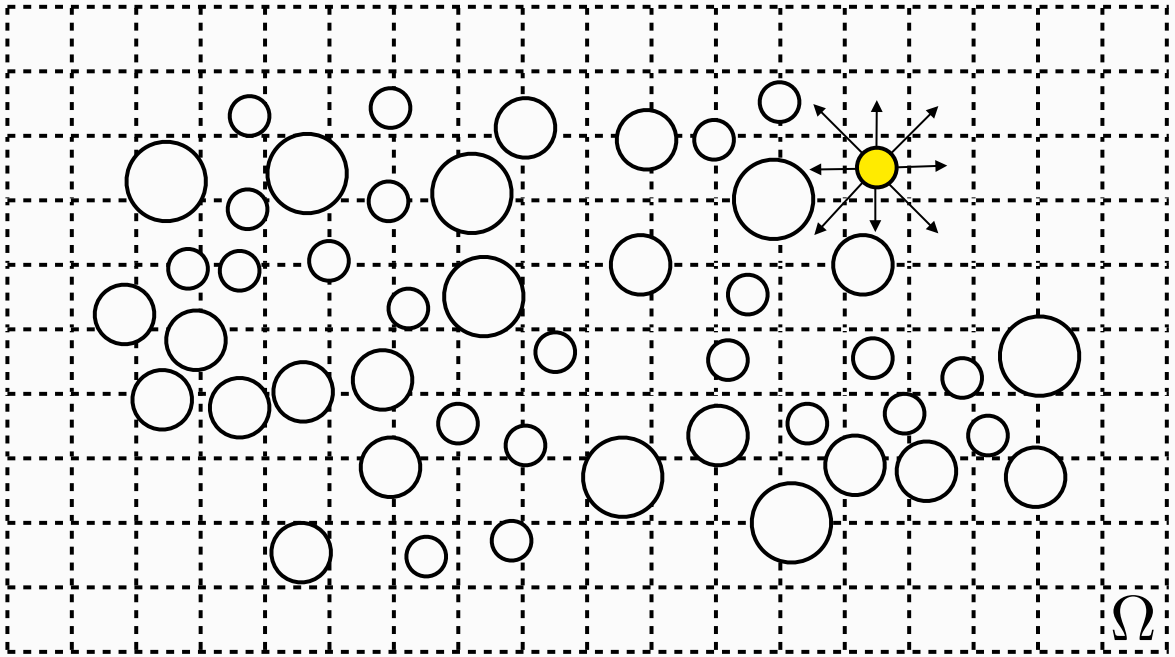
2. For each body, store the identifier in the square boxes that intersect with the current body boundary box



IMPROVEMENTS

Bucket sort

3. Restrict the movement of each body to one box at each time step

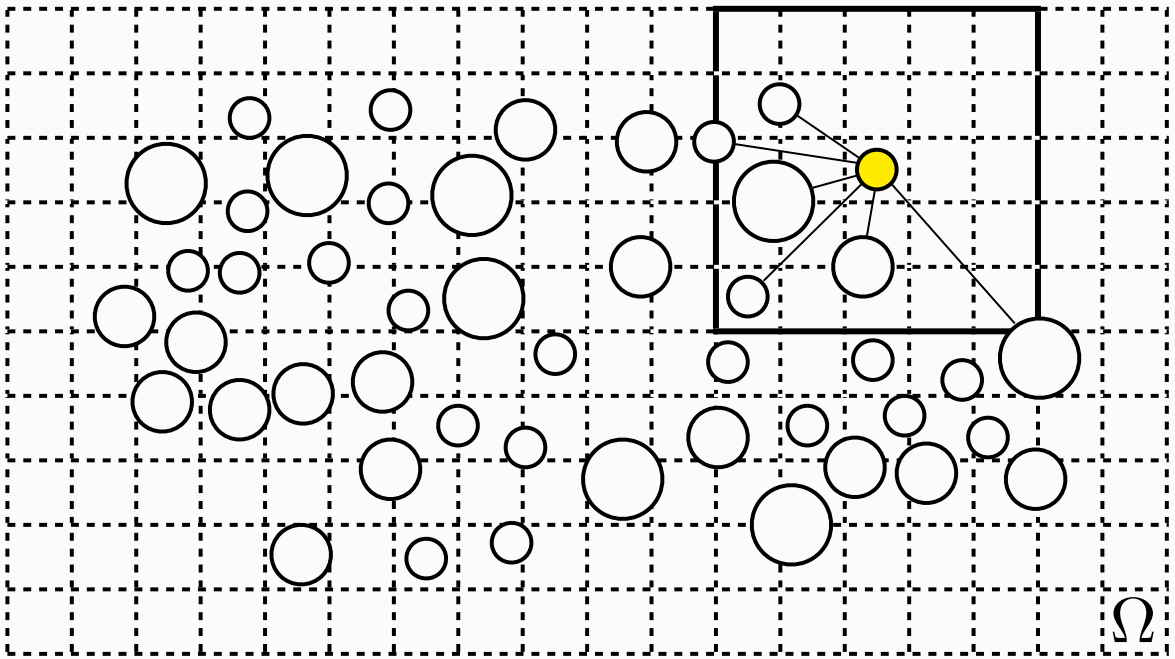


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

IMPROVEMENTS

Bucket sort

4. **Estimating the time of collision**. For each body, consider only the bodies located until the first two levels of neighbor boxes.

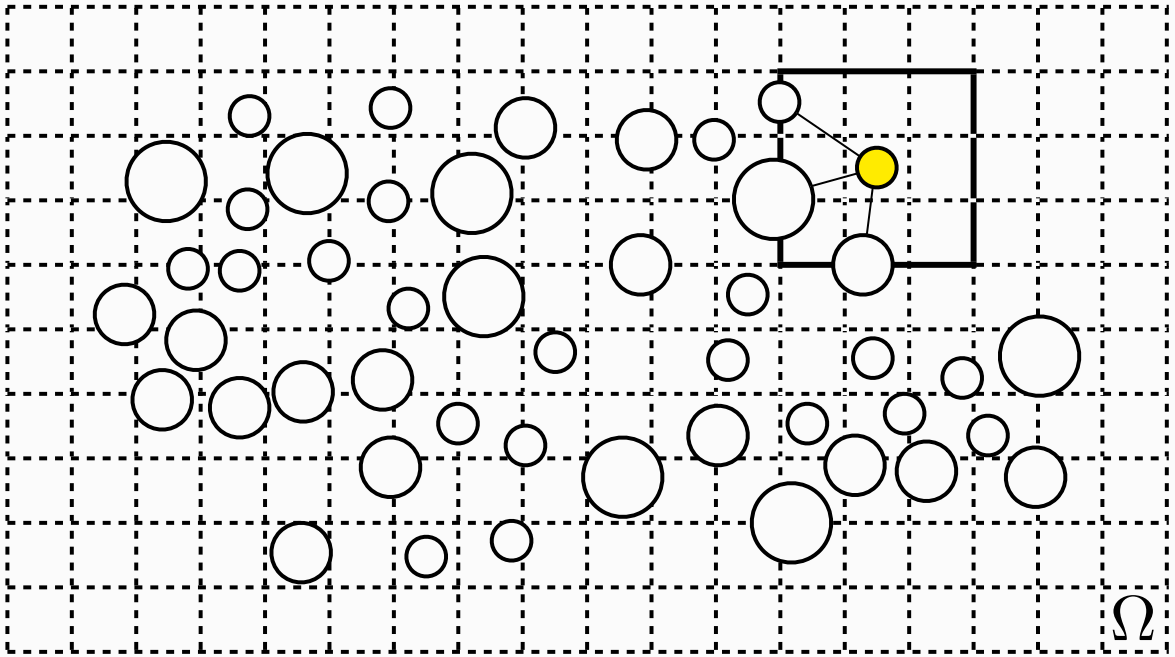


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

IMPROVEMENTS

Bucket sort

5. **Checking intersections**. For each body considers only the bodies located until the first level of neighbor boxes.



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

FIRST ALGORITHM

Closest points between two bodies A and B.

- for each node of the surface mesh of A do
  - for each face of the surface mesh of B do
    - determine the point inside the face of B closest to node of A
    - if the distance between this point and the node of A is the shortest obtained until now then
      - store this point and the node of A as the closest points
    - end if
  - end for
- end for

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

FIRST ALGORITHM

Closest points between two bodies A and B.

- for each node of the surface mesh of B do
  - for each face of the surface mesh of A do
    - determine the point inside the face of A closest to node of B
    - if the distance between this point and the node of B is the shortest obtained until now then
      - store this point and the node of B as the closest points
    - end if
  - end for
- end for

FIRST ALGORITHM

Closest points between two bodies A and B.

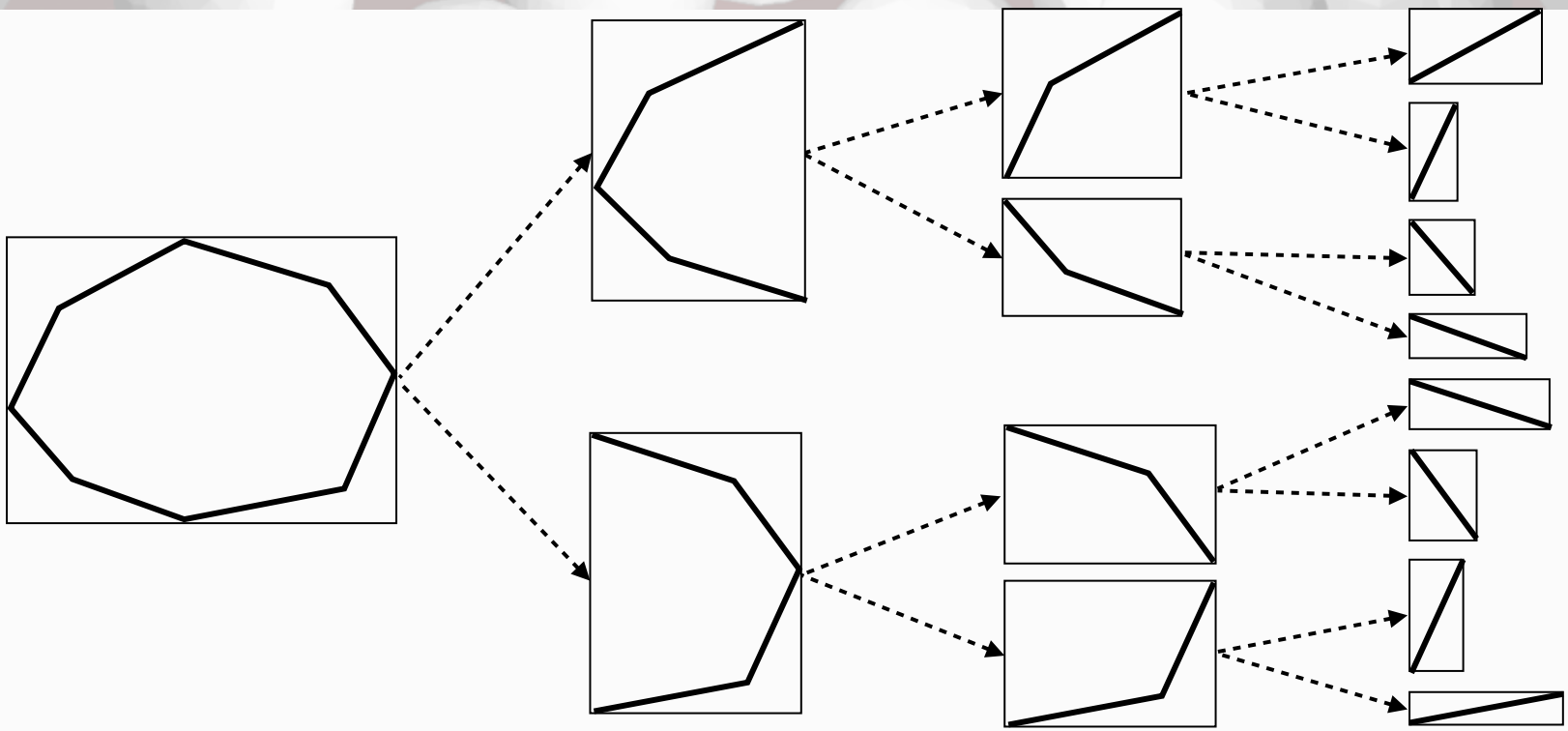
- for each edge of the surface mesh of A do
  - for each edge of the surface mesh of B do
    - determine the closest points between the edges of A and B
    - if the distance between these points is the shortest obtained until now then
      - store these points as the closest points
    - end if
  - end for
- end for

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

IMPROVEMENTS

SKD-Trees

These structures drastically reduce the use of faces in a body to find the shortest distance with a given point.

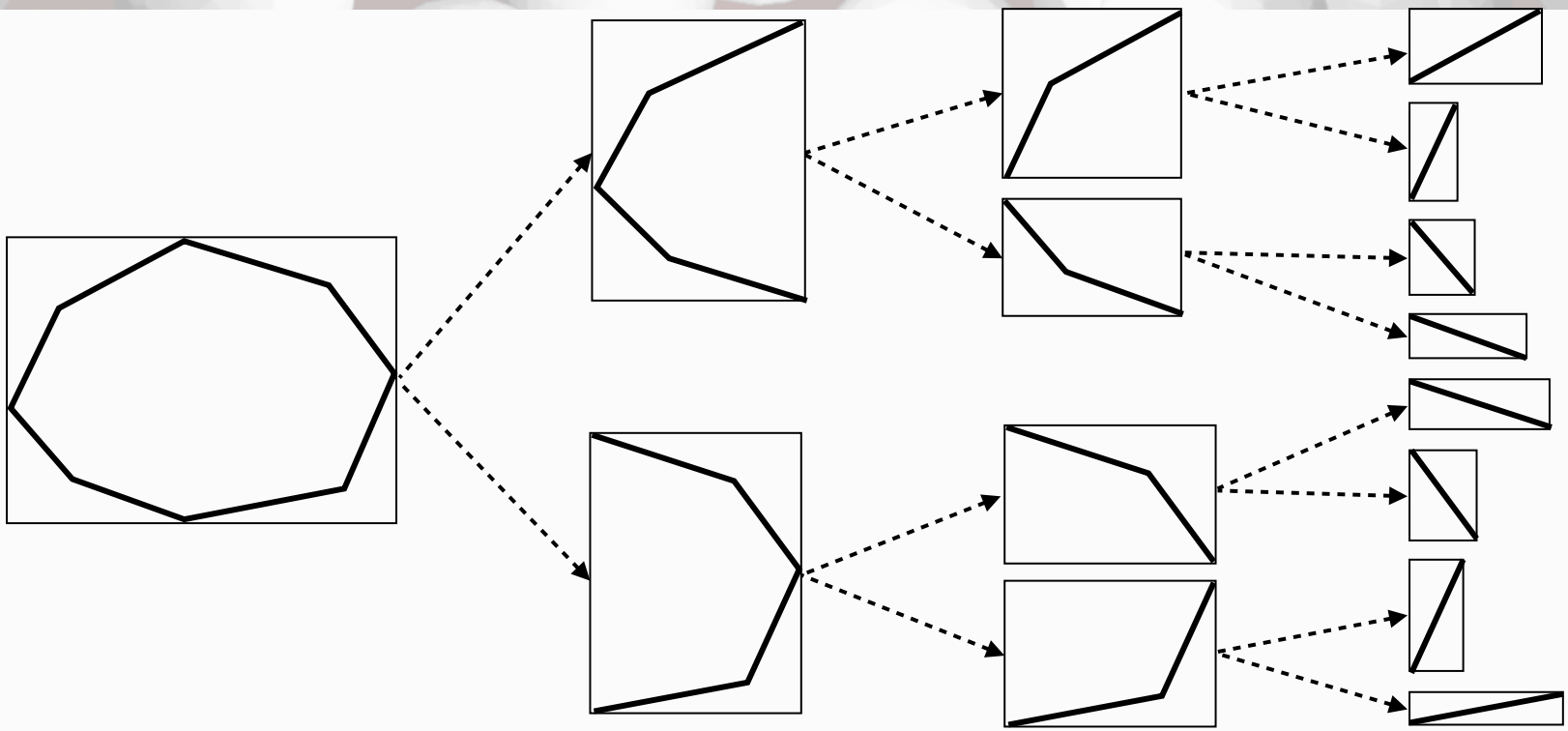


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

IMPROVEMENTS

SKD-Trees building

Each node in these binary trees has associated a boundary box of a set of faces of the particle.

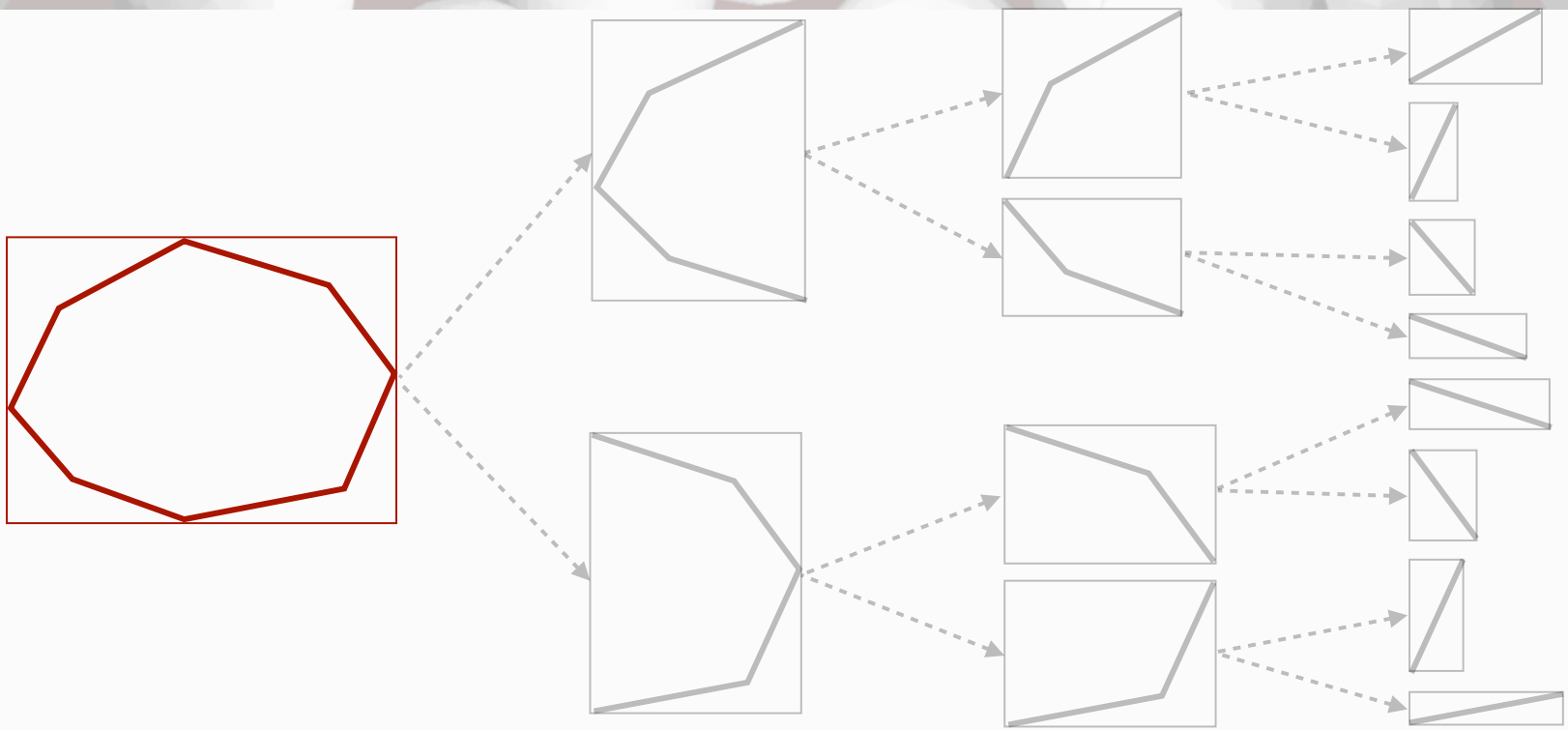


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

IMPROVEMENTS

SKD-Trees building

The root node has associated the boundary box of all the faces of the particle.

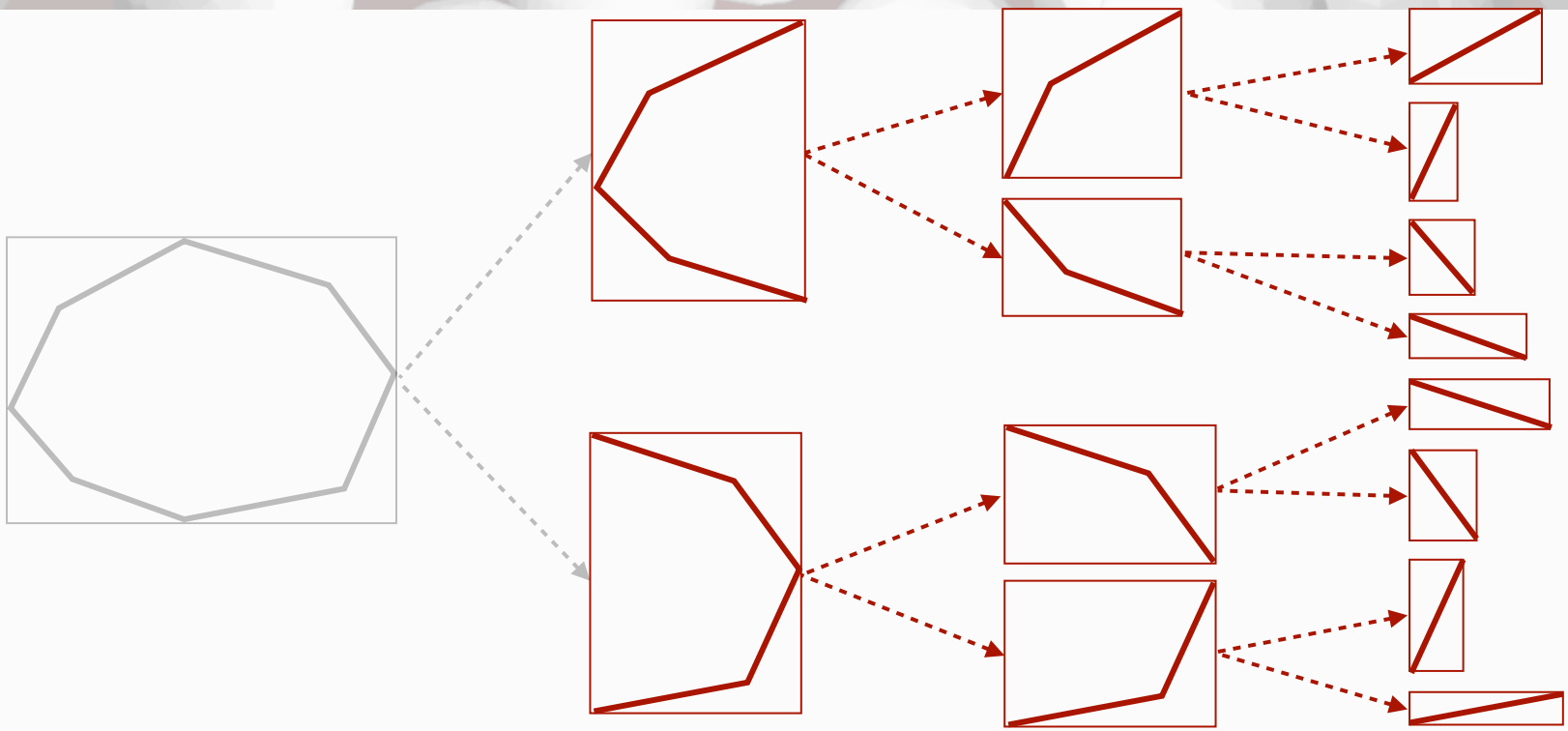


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

IMPROVEMENTS

SKD-Trees building

Each child node has associated the boundary box of the half of the faces of its father node.

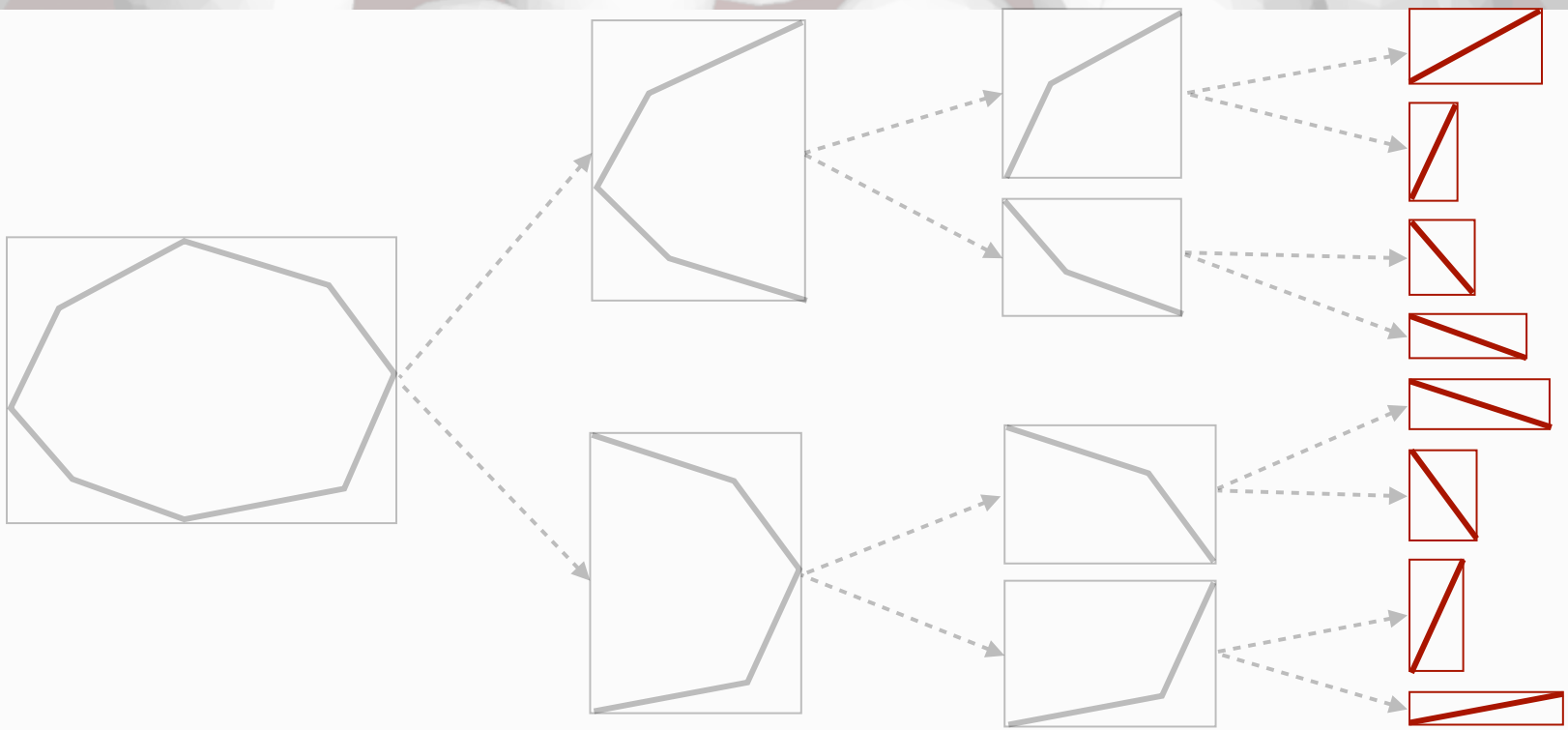


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

IMPROVEMENTS

SKD-Trees building

Only the leaf node include also the face identifier.

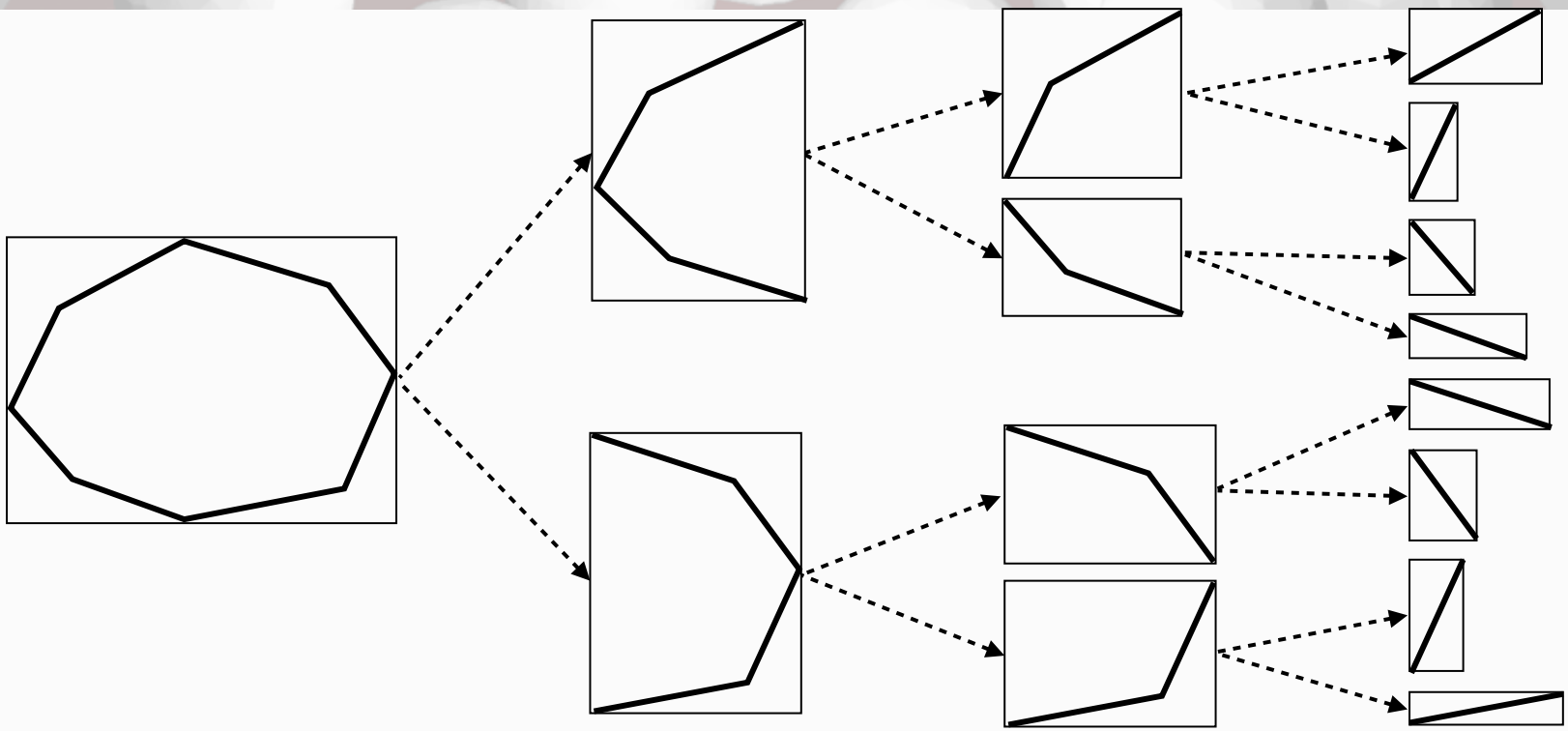


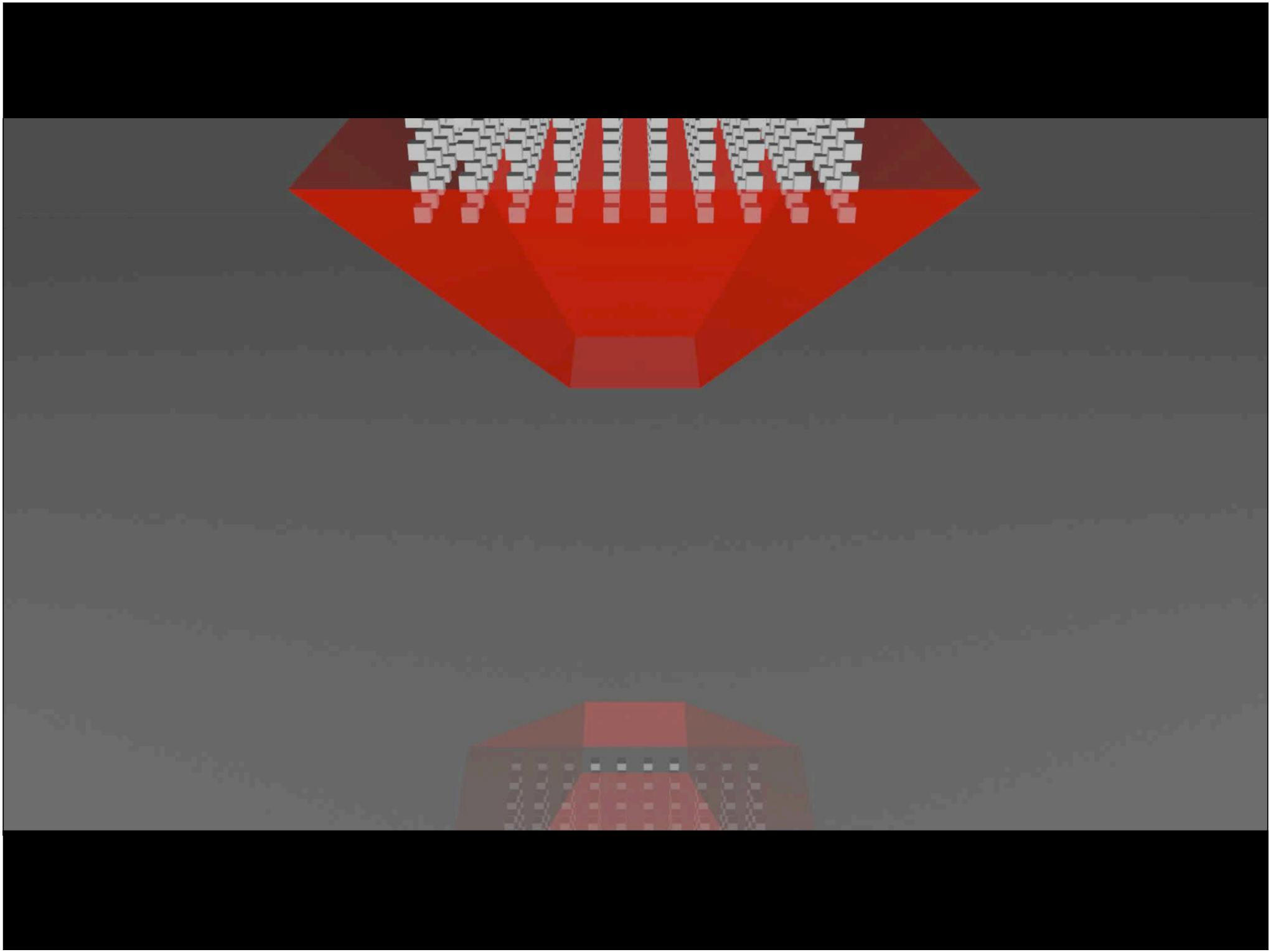
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

IMPROVEMENTS

SKD-Trees

Idea. The use of skd-trees allows to check many more boundary boxes than faces for finding the shortest distance.





Barcelona Supercomputing Center
Centro Nacional de Supercomputación

THE NEWTON EQUATIONS

Implementation. The linear quantities

Consider the force exerted $\mathbf{f}_{n+1}$ at a given time t_{n+1} .Then,

$$\mathbf{a}_{n+1} = \frac{\mathbf{f}_{n+1}}{m}$$

Integrating numerically by using Newmark

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \Delta t(1-\gamma)\mathbf{a}_n + \Delta t\gamma\mathbf{a}_{n+1}$$

$$\mathbf{u}_{n+1} = \mathbf{u}_{n+1} + \Delta t\mathbf{v}_n + \Delta t^2(1/2-\beta)\mathbf{a}_n + \Delta t^2 + \beta\mathbf{a}_{n+1}$$

THE EULER EQUATIONS

The angular quantities

The angular acceleration is equal to

$$\boldsymbol{\alpha}(t) = \mathbf{I}^{-1}(t) \cdot [\boldsymbol{\tau}(t) - \boldsymbol{\omega}(t) \times (\mathbf{I}(t) \cdot \boldsymbol{\omega}(t))]$$

where the inertia tensor is usually determine by

$$\mathbf{I}(t) = \mathbf{R}(t) \cdot \mathbf{J} \cdot \mathbf{R}^T(t)$$

and

$$[\dot{\mathbf{R}}(t) \cdot \mathbf{R}^T(t)] = [\mathbf{W}(t)] = \begin{bmatrix} 0 & -\omega_3(t) & \omega_2(t) \\ \omega_3(t) & 0 & -\omega_1(t) \\ -\omega_2(t) & \omega_1(t) & 0 \end{bmatrix}$$

THE NEWTON-EULER EQUATIONS

Implementation. The angular quantities

Consider the torque exerted $\boldsymbol{\tau}_{n+1}$ at a given time t_{n+1}. Then,

$$\mathbf{R}^{i+1} = \mathbf{R}_n + \Delta t \mathbf{W}^n \cdot \mathbf{R}_n$$

$$(\mathbf{I}^{-1})_{n+1} = (\mathbf{R}^T)_{n+1} \cdot \mathbf{J}^{-1} \cdot \mathbf{R}_{n+1}$$

$$\boldsymbol{\alpha}_{n+1} = (\mathbf{I}^{-1})_{n+1} \cdot [\boldsymbol{\tau}_{n+1} - \boldsymbol{\omega}_n \times (\mathbf{I}_{n+1} \cdot \boldsymbol{\omega}_n)]$$

Integrating numerically by using Newmark

$$\boldsymbol{\omega}_{n+1} = \boldsymbol{\omega}_n + \Delta t(1-\gamma)\boldsymbol{\alpha}_n + \Delta t \gamma \boldsymbol{\alpha}_{n+1}$$

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

THE NEWTON-EULER EQUATIONS

Implementation. Improvement

Iterate until some error be higher than a given tolerance

$$\mathbf{R}_{n+1}^{i+1} = \mathbf{R}_n + \Delta t \mathbf{W}_{n+1}^{i} \cdot \mathbf{R}_{n+1}^{i}$$

$$(\mathbf{I}^{-1})_{n+1} = (\mathbf{R}^T)_{n+1}^{i+1} \cdot \mathbf{J}^{-1} \cdot \mathbf{R}_{n+1}^{i+1}$$

$$\boldsymbol{\alpha}_{n+1}^{i+1} = (\mathbf{I}^{-1})_{n+1} \cdot [\boldsymbol{\tau}_{n+1} - \boldsymbol{\omega}_{n+1}^{i} \times (\mathbf{I}_{n+1} \cdot \boldsymbol{\omega}_{n+1}^{i})]$$

Integrating numerically by using Newmark

$$\boldsymbol{\omega}_{n+1}^{i+1} = \boldsymbol{\omega}_n + \Delta t(1-\gamma)\boldsymbol{\alpha}_n + \Delta t \gamma \boldsymbol{\alpha}_{n+1}^{i+1}$$

THE NEWTON-EULER EQUATIONS

Implementation. More improvements

Numerical errors will appear in the coefficients of $\mathbf{R}(t)$ so that the rotation matrix will no longer an orthogonalized matrix.

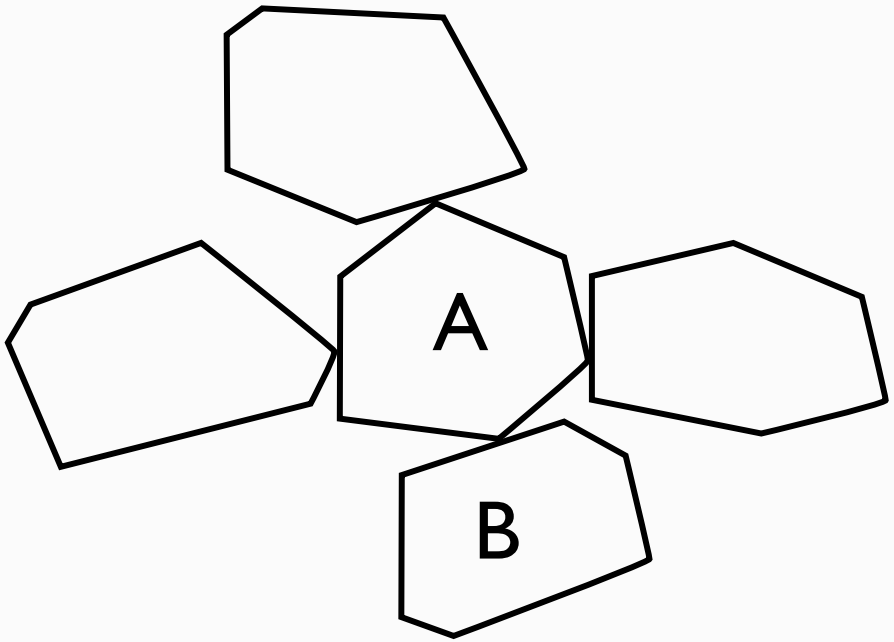
Solution. Use unit **quarternions** to represent rotations.

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

MORE IMPROVEMENTS

Distance correction.

Any group of bodies in contact has to be a minimum distance of separation.

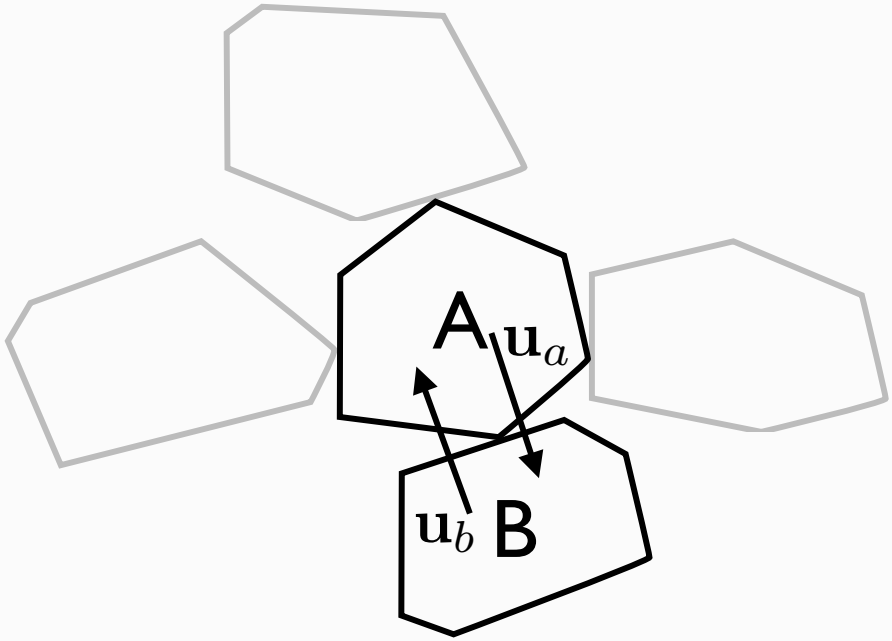


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

MORE IMPROVEMENTS

Distance correction.

Consider the estimation of the time of collision between the bodies A and B if its separation is equal to zero.

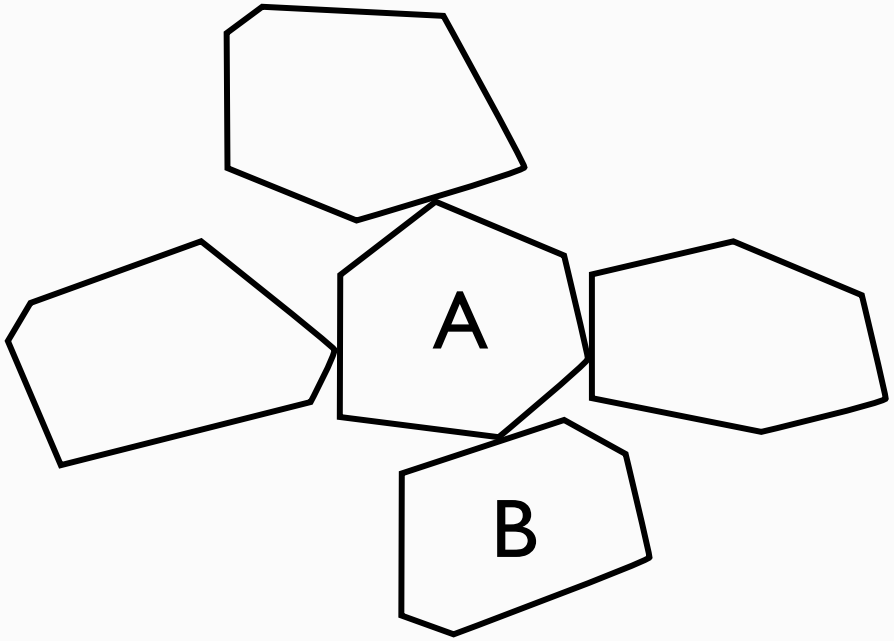


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

MORE IMPROVEMENTS

Distance correction.

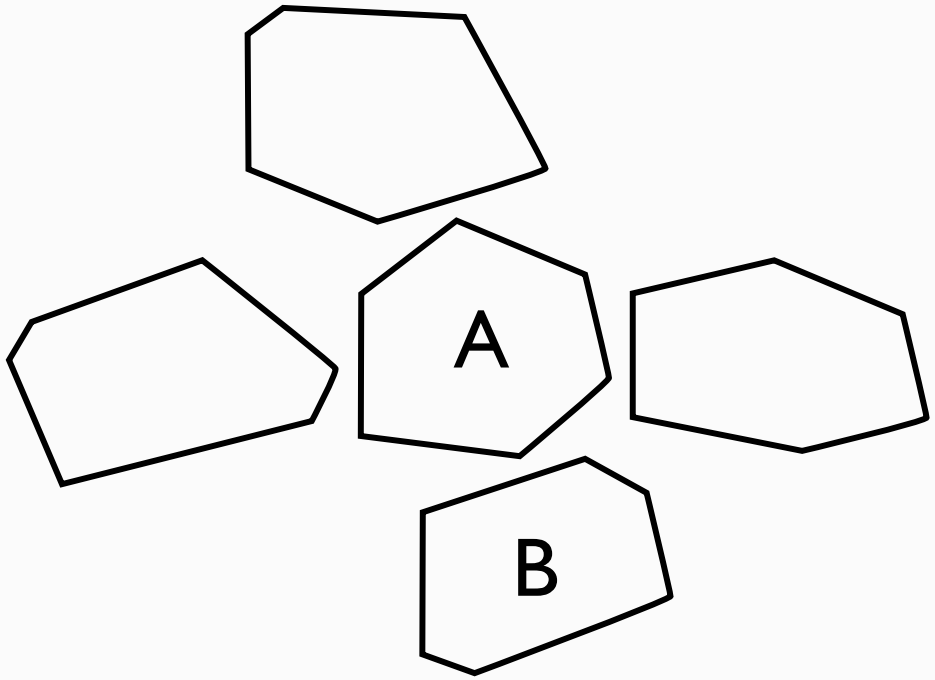
This time will be equal to zero. As a consequence, the simulation will be not able to advance.



MORE IMPROVEMENTS

Distance correction.

Solution. Keep a minimum distance between all the bodies (a very small distance)

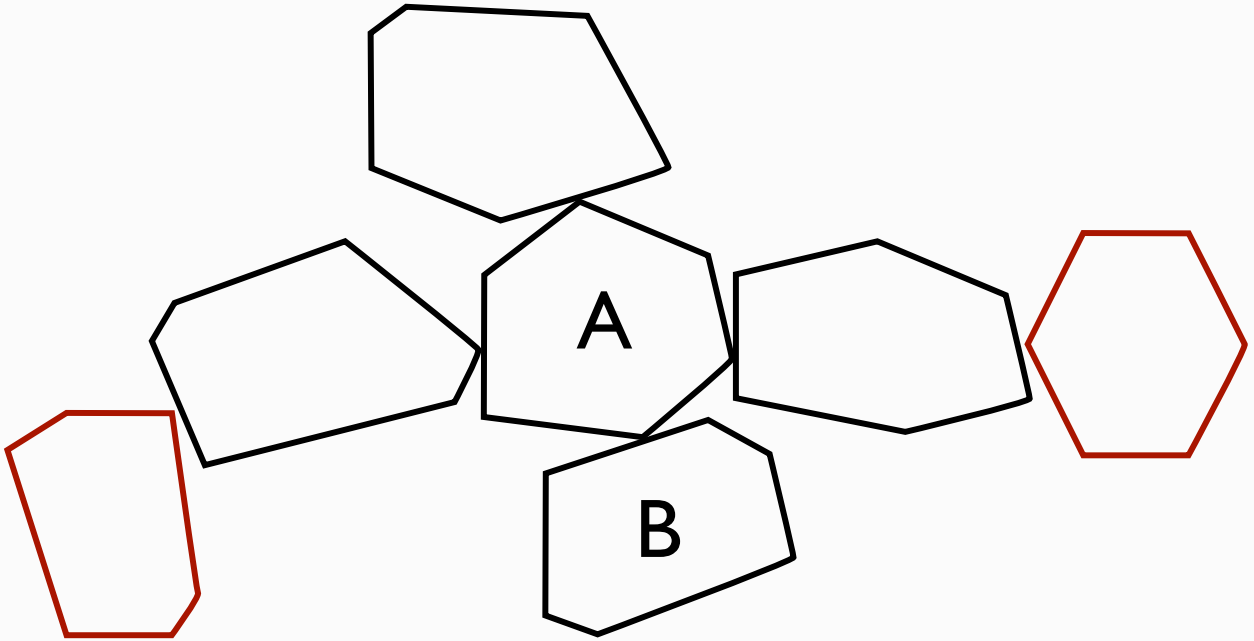


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

MORE IMPROVEMENTS

Distance correction.

Problem. The distance correction of a body can cause interpenetrations between other bodies.

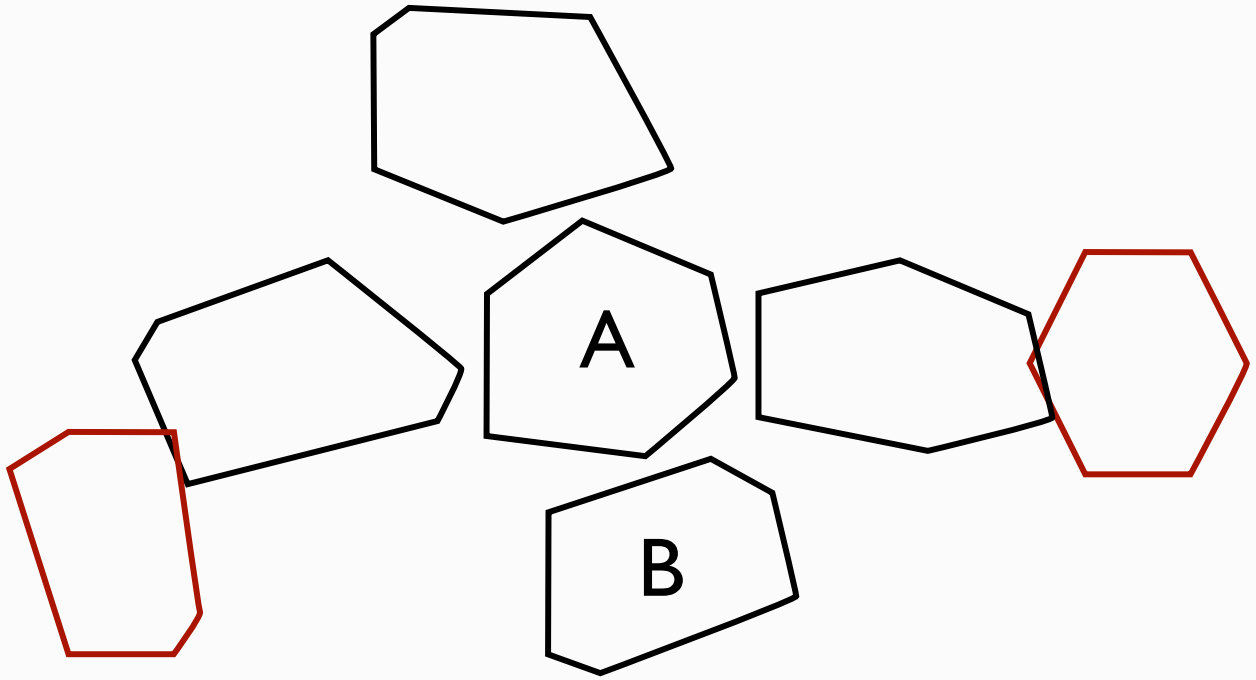


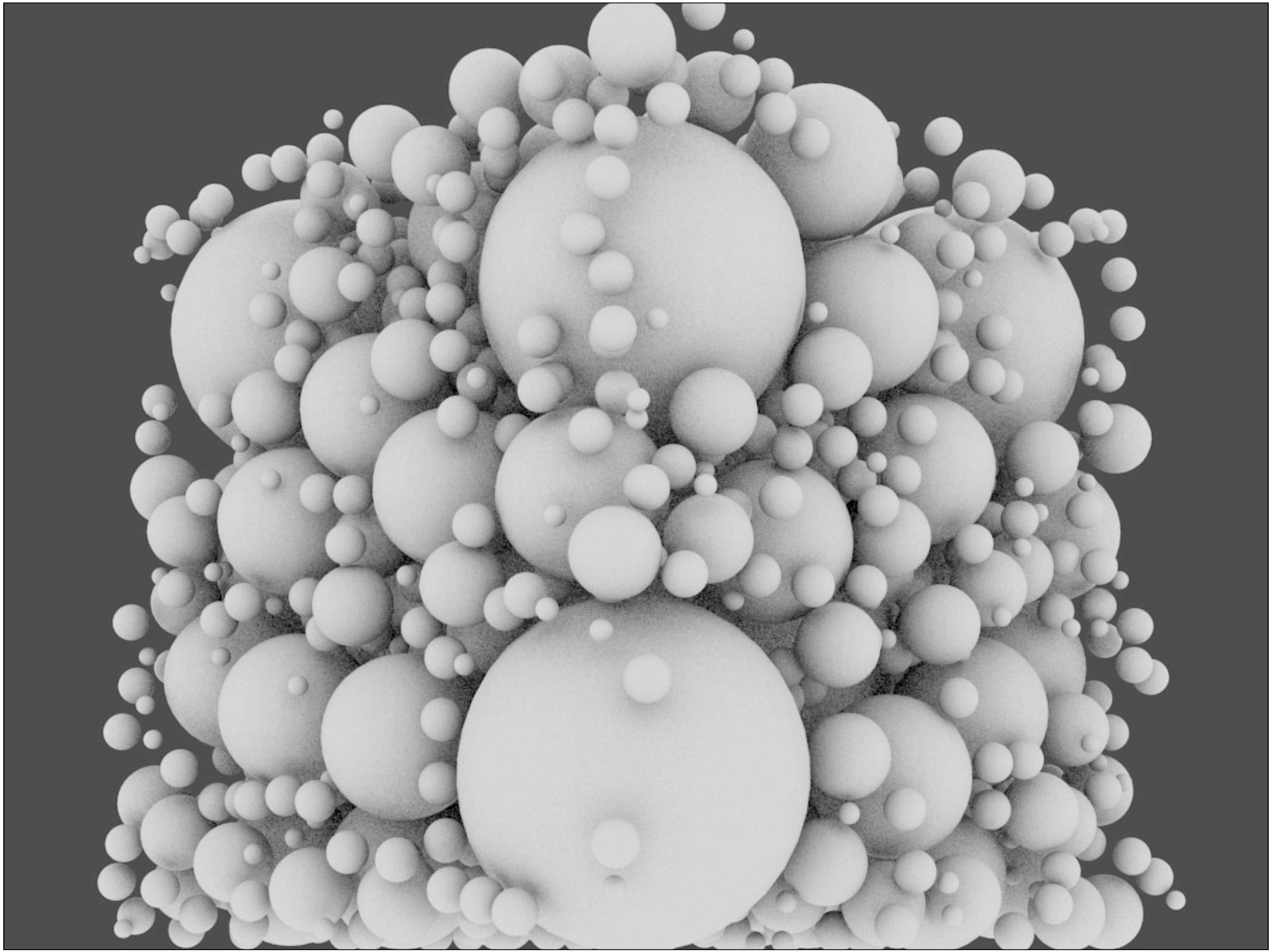
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

MORE IMPROVEMENTS

Distance correction.

Problem. The distance correction of a body can cause interpenetrations between other bodies.

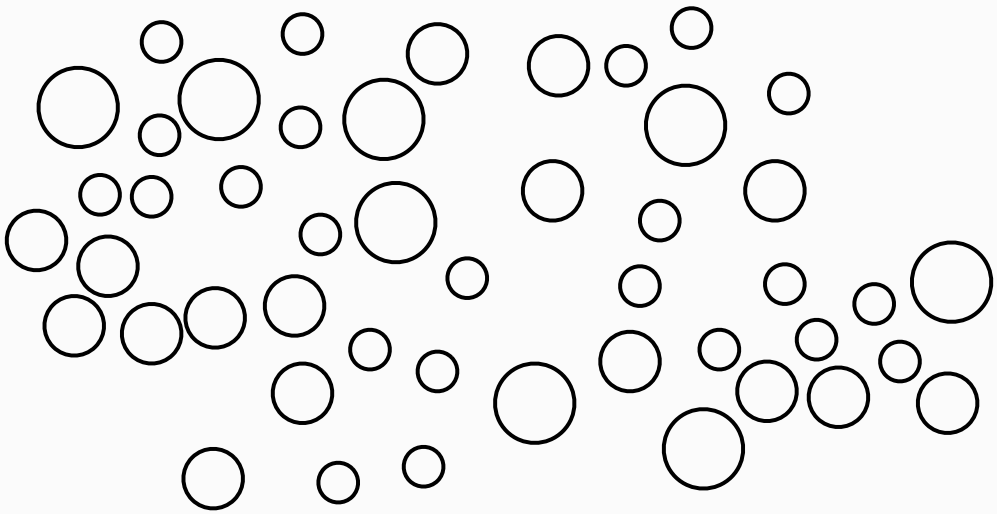




MORE IMPROVEMENTS

PARALLEL ISSUES

Consider again the following group of bodies.

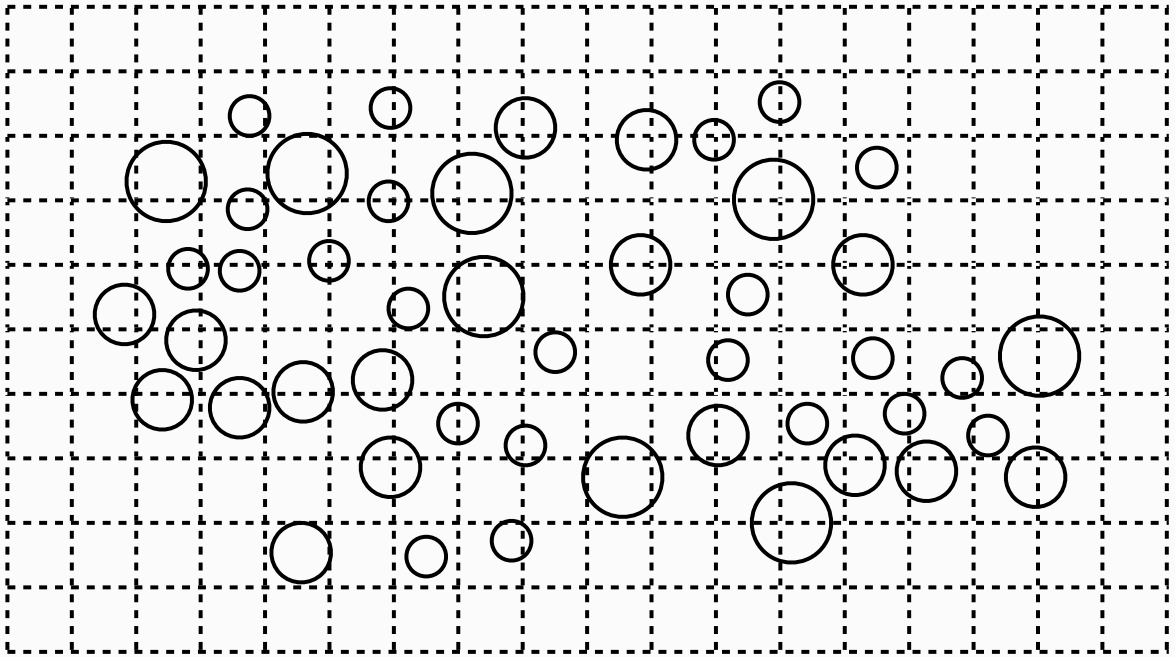


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

MORE IMPROVEMENTS

PARALLEL ISSUES

The bucket sort divides the domain in boxes

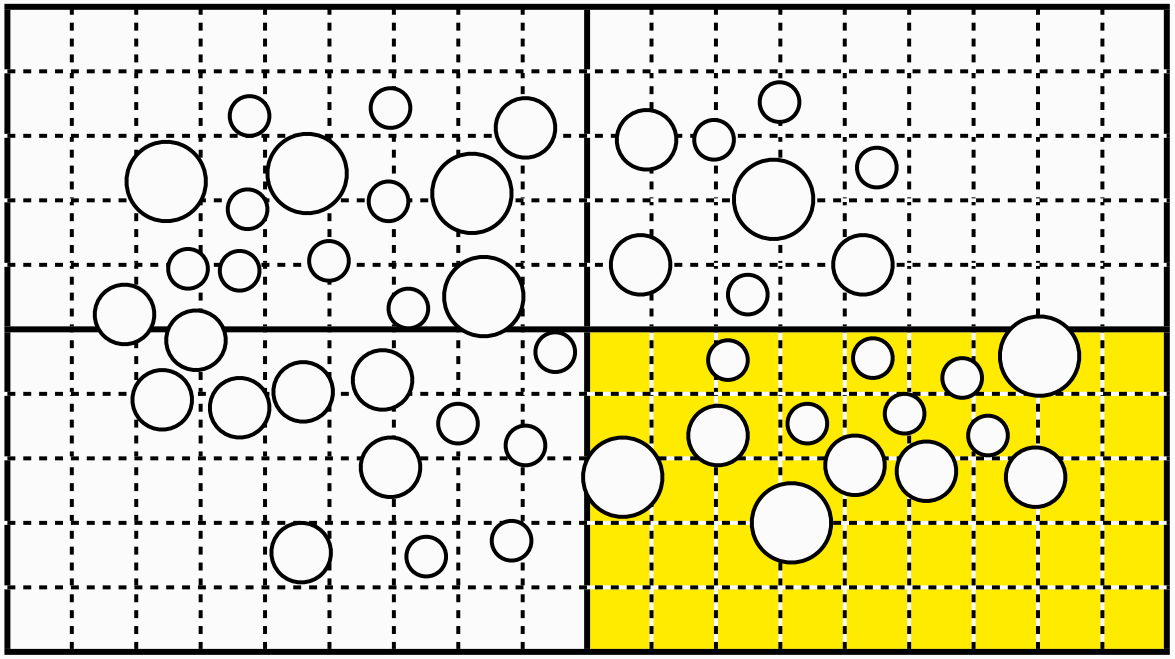


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

MORE IMPROVEMENTS

PARALLEL ISSUES

Idea. The data of the domain is again divided to group the square boxes in subdomains.

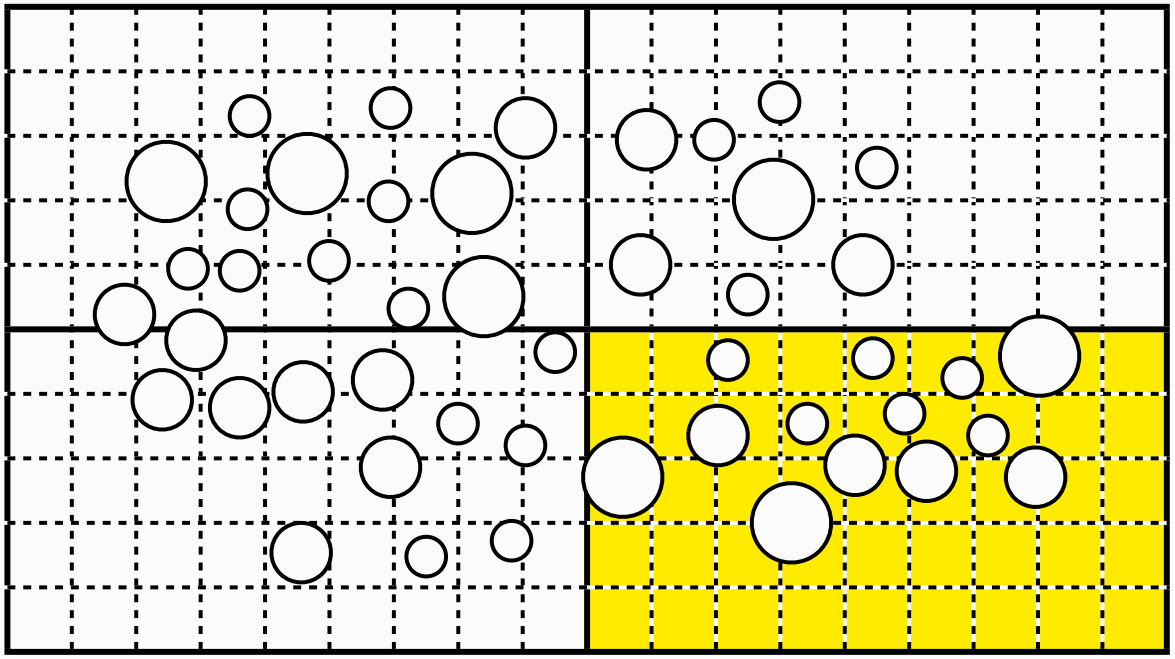


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

MORE IMPROVEMENTS

PARALLEL ISSUES

For each body, store the corresponding data in the subdomains that intersect with the current boundary box.

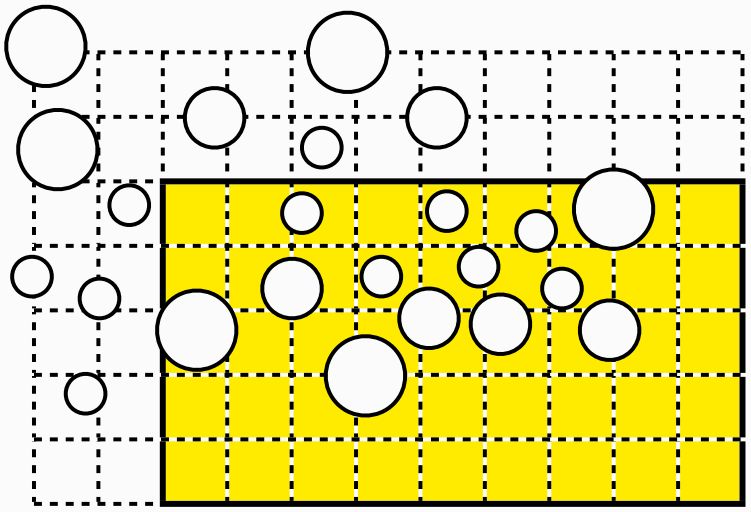


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

MORE IMPROVEMENTS

PARALLEL ISSUES

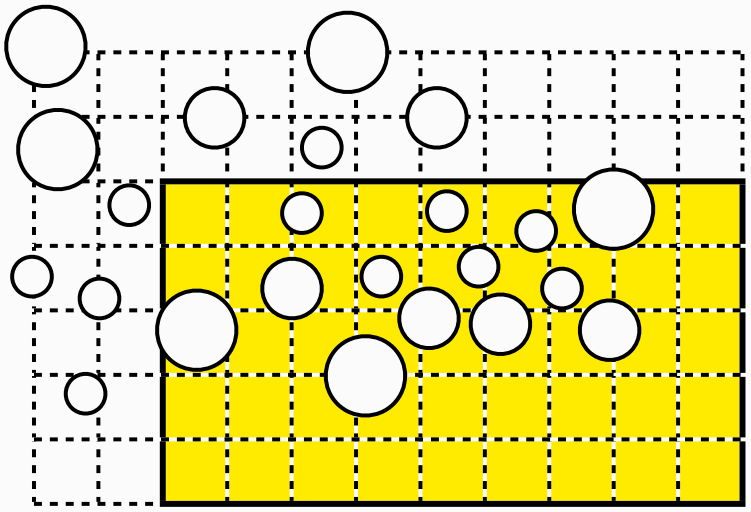
Remember. For each body, it is necessary to consider the bodies located until the first two levels of neighbor boxes

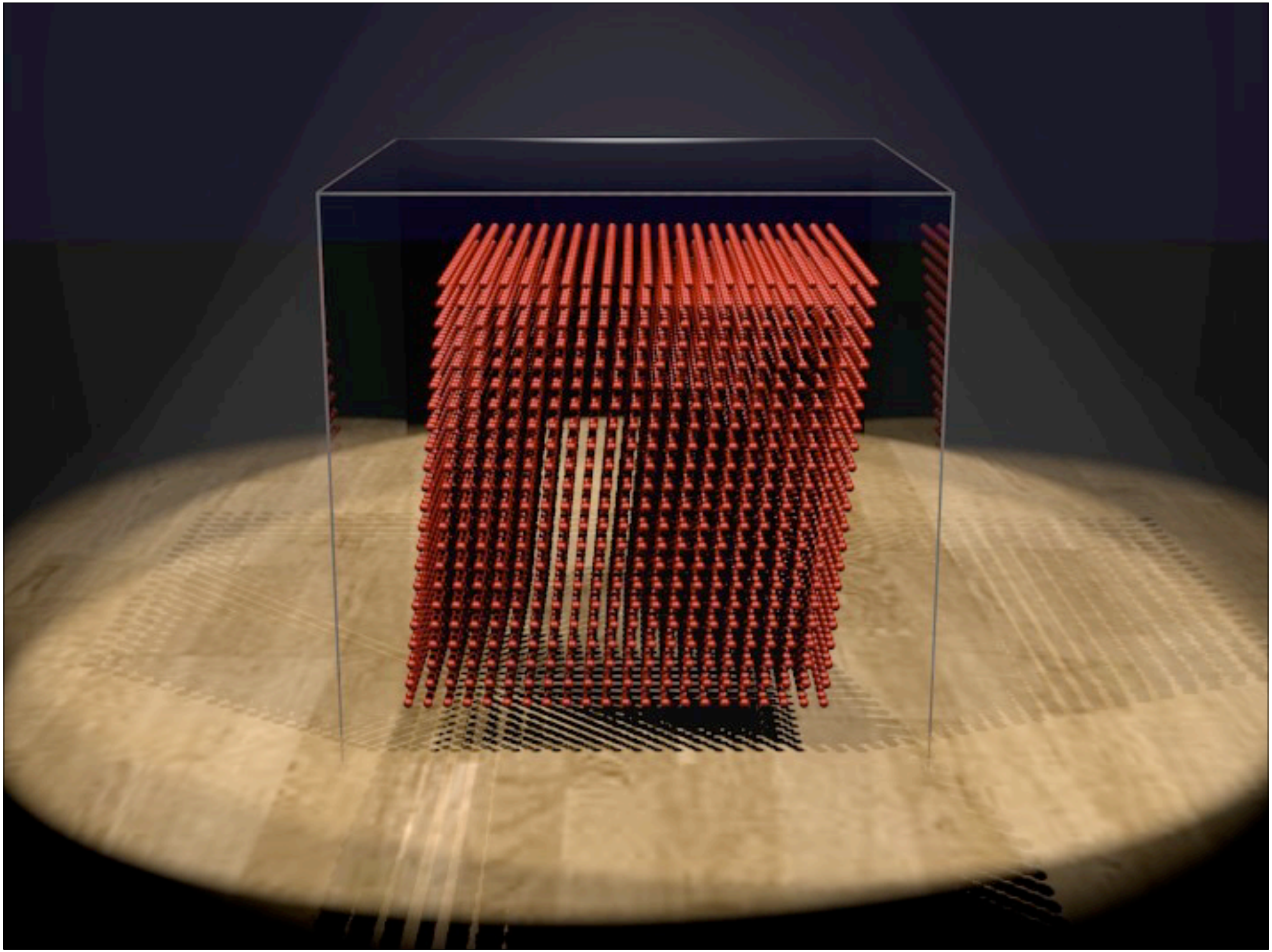


MORE IMPROVEMENTS

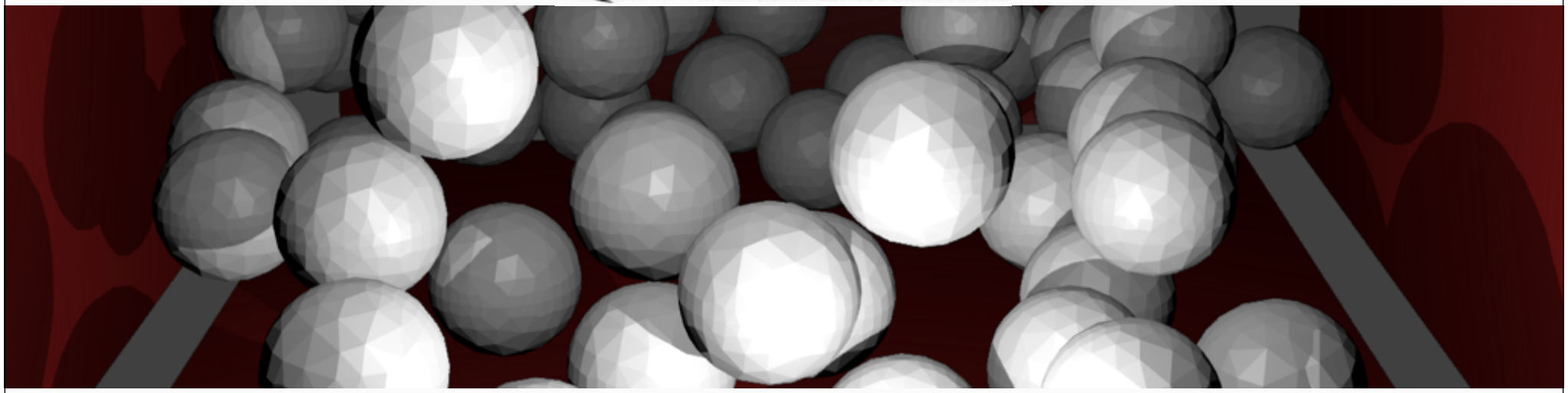
PARALLEL ISSUES

Then, to reduce the interchange of data, each subdomain adds the data of the first two levels of neighbor square boxes that are stored in other subdomains.





Barcelona Supercomputing Center
Centro Nacional de Supercomputación



THANKS FOR YOUR ATTENTION!